1991

Theoretical study of information capacity of Hopfield neural network and its application to expert database system

Kesig Lee
Iowa State University

Order Number 9126214

Theoretical study of information capacity of Hopfield neural network and its application to expert database system

Lee, Kesig, Ph.D.

Iowa State University, 1991

U·M·I
300 N. Zeeb Rd.
Ann Arbor, MI 48106

Theoretical study of information capacity of Hopfield neural network and its application to expert database system

by

Kesig Lee

A Dissertation Submitted to the
Graduate Faculty in Partial Fulfillment of the
Requirements for the Degree of
DOCTOR OF PHILOSOPHY

Major: Computer Science

Approved:

Signature was redacted for privacy.

In Charge of Major Work

Signature was redacted for privacy.

For the Major Department

Signature was redacted for privacy.

For the Graduate College

Members of the Committee:

Signature was redacted for privacy.

Iowa State University
Ames, Iowa
1991

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENT

I am deeply grateful to my advisor, Dr. S. C. Kothari for his valuable advice and support. I would also like to thank Dr. D. Fernandez-Bach, Dr. W. Kuo, Dr. J. Wong, Dr. C. T. Wright, and Dr. D. Shin for their time and energy.

Special thanks go to **Samsung** family for their continuous support from the beginning of my study until now; especially to my wife, Inhee, and daughters, Moonyoung and Minsun for their sacrifice. I will always remember the love and encouragement they have given me.

Last, I cannot but thank Dr. L. Miller who has never hesitated to discuss my personal life problems.

1. INTRODUCTION

A conventional computer system can solve complex mathematical problems very fast, yet it can't efficiently process high-level intelligent functions of human brain such as pattern recognition, categorization, and associative memory. Human beings can easily recognize their friends whom they met several years ago by recalling the pertinent information about the friends. But no programs yet can compete with human beings in general-purpose pattern recognition.

In fact, neurons which are basic elements of human brain have computing time of tens of milliseconds, which means that they are five or six orders of magnitude slower than conventional silicon logic gates [8, 18]. How can then human beings easily do the high-level intelligent functions with so slow neurons? For several decades, it has been a goal of science and engineering to answer the question and to develop an intelligent machine.

There have been two philosophically different research activities for pursuing the goal. One is *Artificial Intelligence* (*Symbolic, Macroscopic*) approach and the other is *Neurological* (*Biological, Microscopic*) approach [85, 91]. Researchers following the former approach have tried to manipulate the human mind's symbolic representation of the external world. To do that, they need to create an explicit programming structure as an internal representation of the external world and need to design a

series of operations handling the structure algorithmically [85, 91]. Unfortunately, so far researchers have failed in providing an useful, and flexible enough framework for executing high-level intelligent functions.

On the other hand, researchers following the latter approach have tried to model the brain itself. Because of reasonable success of Artificial Intelligence (AI) approach using conventional digital computers in 1970s and also because of Minsky and Papert's [70] theoretical criticism, the neurological approach was abandoned until 1980. But thanks to new insights by mathematical theories, the revolution in computer technology, and better understanding of neurobiological process of human brain [18], the neurological approach was resurrected around 1980. Subsequently, interest in neurological approach as an alternative to conventional computing has tremendously increased. The neurological approach can be divided into the following two categories [91].

- **Biological Modeling:** Study of human brain itself on the view of the structure and function.
- **Technological Modeling:** Study of artificial neural network methodologies for simulating the structure and function of human brain.

1.1 Motivation and Scope of This Research Work

A neural network is a computational structure for modeling some of the high-level intelligent functions of human brain. Recently, neural networks have attracted considerable attentions as a novel computational system because of the following expected benefits which are generic characteristics of human brain [18].

- High processing speed through massive parallelism.
- Learning as a means of efficient knowledge acquisition.
- Robustness arising from distributed information processing.

Neural networks are being studied from a different point of view in many disciplines such as psychology, mathematics, statistics, physics, engineering, computer science, neuroscience, biology, and linguistics. Depending on disciplines, neural networks have diverse nomenclature as *artificial neural networks, connectionism, PDPs, adaptive systems, adaptive networks*, and *neurocomputers*. This presents the following problems to a researcher.

1. Difficulty of finding the starting point of a research in the current state of the art.
2. Difficulty of finding possible theoretical issues.
3. Difficulty of finding real applicable areas from known theoretical results.

It is, of course, impossible to provide standardized general answers for the above problems. We try to answer the problems from the viewpoint of computer scientists. The objectives of this research work are:

1. Providing a global picture of the current state of the art by surveying a score of neural networks chronologically and functionally,
2. Providing a theoretical justification for well-known empirical results about the information capacity of the Hopfield neural network, and

3. Providing an experimental logical database system using Hopfield neural network as an inference engine.

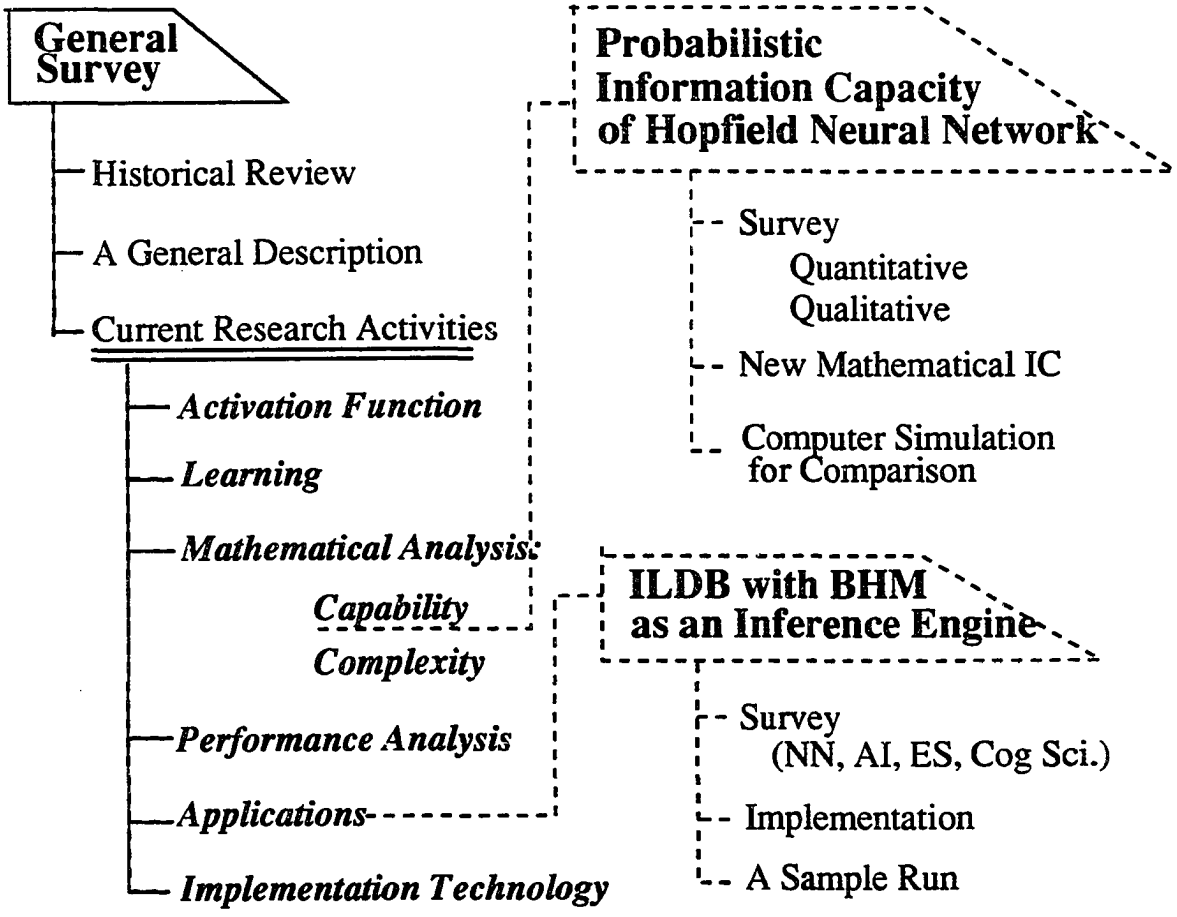

Figure 1.1: Thesis organization

2. SURVEY OF NEURAL NETWORKS

As we mentioned before, the study of neural networks is a field that is built upon studies in diverse disciplines. Hence, it is very difficult to provide a systematic guide to the background of concepts employeed in this field. In our survey, we cover the following:

1. A brief and chronological review of important milestones in the history of neural networks, which helps to clarify the origins of ideas for various research activities in neural networks,

2. A description of common paradigms that exist in different neural networks in spite of the apparent diversity,

3. A grouping of current research topics according to their themes.

2.1 Historical Review of Neural Networks

The history of neural networks is complicated and there are many contributors. We try to trace the milestones in the main stream of the neural network studies. For more detailed history, we refer the reader to [8], [70], and [79].

2.1.1 The 1940s: fundamental concepts

McCulloch and Pitts (Boolean Logic) — 1943: They showed that an arbitrary logical function can be executed by a neural network of interconnected digital neurons. The threshold function introduced by them was used in Hopfield neural network [43, 44] and Kosko's Bidirectional Associative Memory [58, 59]. Unfortunately, no notion of *learning* was imbedded in their network.

Hebb (Synaptic Learning Rule) — 1949: From the observation that neural networks might learn by constructing internal representation of concepts, he suggested a learning paradigm that is called *Hebbian learning.*

2.1.2 The 1950s: the concepts of learning

Minsky — 1951: He designed the first real learning system based on Hebbian learning.

Rosenblatt (Perceptron) — 1957: He generalized the McCulloch and Pitts' network by adding the notion of learning. Unfortunately, he couldn't suggest a learning rule for 3-layer perceptrons. This limitation was criticized by Minsky in 1969.

Widrow (Adaline) — 1959: He proved that the error between the target patterns and the output patterns will find a global minimum under certain conditions. Unfortunately, he also failed in suggesting a learning rule for 3-layer Adalines.

2.1.3 The 1960s: theoretical foundations

Grossberg — 1964: He tied psychological processes and biological processes into unified theories. His initial studies led to the development of *instar*, *outstar*, and *avalanche learning* rules in 1974.

Amari — 1967: One of the most influential researchers of neural neural network theory. He tried to combine biological neural network activity and rigorous mathematical expertise. He provided the mathematical foundation to describe the dynamics of randomly connected neural networks.

Anderson (Linear Associative Memory) — 1968: He pioneered the study of neural networks for associative memory. A linear associative memory was introduced in 1972.

Minsky again — 1969: He showed all the limitations of 2-layer perceptrons especially by showing that 2-layer perceptrons will only work for problems with linearly separable solution spaces.

2.1.4 The 1970s: wilderness years

After Minsky's criticism against perceptron-like neural networks, the majority of neural network funding was reallocated into AI programs. The representation of knowledge in the field of AI became the most important research activity. At this time, many new and powerful ideas in AI were developed, for example, *conceptual dependency, production system, relational database, scripts, semantic network, non-monotonic logic*, etc., [70].

Despite a lack of research funding, several dedicated researchers tried to develop variety of mathematical theories. Their efforts eventually became the fertilizer for the resurgence of neural networks after 1980. Representative researchers and their main studies are summarized as follows:

Anderson: Linear Associative Memory,

Fukushima: Cognitron,

Grossberg: Competitive-Cooperative Learning,

Kohonen: Correlation Matrices as Associative Memories, and

Sejnowski: Neurological Evidence for Covariance Learning Rule.

2.1.5 1980 — : resurgence

The invention of the backpropagation algorithm [76] has played the most important role in the resurgence of neural networks. Despite its limitation, it provides a mathematical foundation by suggesting a systematic method for training multi-layer neural networks. Representative research activities are as follows:

Anderson: Brain-State-in-a-Box (BSB),

Feldman and Ballard: Connectionism,

Fukushima: Neocognitron,

Grossberg and Center for Adaptive Systems: Adaptive Resonance Theory (ART),

Hecht-Nielsen: Counterpropagation Network,

Hopfield/Tank: Hopfield Neural Network,

Kohonen: Self-Organizing Feature Map,

Kosko: Bidirectional Associative Memory (BAM),

McClelland, Rumelhart, and PDP Group: Parallel Distributed Processing (PDP) Model, and

Mead and Conway: Replication of animal nervous systems in electronic circuitry.

The basic concepts behind these neural network models and the comparison among them are well described in [13] and [64].

2.2 A General Description of Neural Networks

In spite of the apparent diverse origins of neural networks, there are general paradigms common among several neural network models. We provide a general description of neural networks based on these paradigms. Even though the description can't be regarded as a standard notation or a representation for different neural networks, it can play the role of a good guide for the journey into the world of diverse neural networks.

Informally, a neural network is a collection of highly-interconnected, simple analog processing elements that mimic biological neurons [46, 71]. The main structural characteristics of a neural network can be described as follows:

- Parallel processing,

- Massive interconnections, and
- Emergent collective computational abilities.

More formally, a neural network of N interconnected neurons can be defined as a mapping function **G** from $\mathbf{A} = [a_i]_{N\times 1}$ into itself, where **A** is an N dimensional vector whose element a_i represents an activation state of the i^{th} neuron. A dynamics on the neural network is defined by the function **G**. This formal definition can be functionally reviewed using the following seven major aspects of a neural network [76]:

1. **A Set of Neurons (Units):** Any neural network model begins with a set of simple processing units called neurons. Let N be the number of neurons and u_i be the i^{th} neuron.

2. **A State of Activation:** A representation of the activation state of a neural network at time t is represented as $\mathbf{A}(t) = [a_i(t)]_{N\times 1}$ which is a vector of N real numbers representing the pattern of activation over the set of neurons.

3. **An Output Function for Each Neuron:** Neurons interact by transmitting a signal to their neighbors. By doing so, they affect the activation states of their neighbors. The strength of the signal from a neuron u_i is determined by the output function f_i using the activation state $a_i(t)$.

4. **A State of Output Values:** The current set of output values is represented by $\mathbf{O} = [o_i(t)]_{N\times 1}$ which is a vector of N real numbers. The relation among the activation state, the output function f_i, and the output state is represented by $o_i(t) \xleftarrow{f_i} a_i(t)$.

5. **A Connection Matrix (Weight matrix):** Neurons are connected to each other. The pattern of this connectivity is represented by a matrix $\mathbf{W} = [w_{i,j}]_{N \times N}$, where $w_{i,j}$ is the strength of connectivity from the neuron u_i to the neuron u_j. This matrix $\mathbf{W}$ stores knowledge or information, more specifically, the correlation among pairs of input and output patterns. How a neural network can build up the matrix $\mathbf{W}$ automatically from the given external information corresponds to the classical and main issue in neural networks, i.e., *learning.*

6. **Rule of Propagation:** This rule is used to take the output values of the neurons and combine them with the weight matrix $\mathbf{W}$ to produce a net input for each neuron. The net input can be represented by the vector product $\mathbf{NET}(t) = \mathbf{W} \times \mathbf{O}(t)$, where $\mathbf{NET}(t) = [net_i(t)]_{N \times 1}$ is a vector of N real numbers.

7. **Activation Rule:** This is the rule for producing a new activation state $\mathbf{A}(t+1)$ from $\mathbf{NET}(t)$ and $\mathbf{A}(t)$. This can be represented by $\mathbf{A}(t+1) \xleftarrow{F} (\mathbf{A}(t), \mathbf{NET}(t))$, where F is generally a differentiable function. There are 3 different types of activation rules.

   Asynchronous Updating: a neuron changes its activation state at each discrete time t.

   Fully Parallel Updating: all neurons change their activation states at the same time.

   Partially Simultaneous Updating: a group of neurons change their activation states simultaneously at each discrete time t.

Remark : Neural network models differ from each other depending on the function F and the learning algorithm.

The relationship between the seven aspects and the operation of a neural network are described in Figure 2.1 and Figure 2.2 respectively.

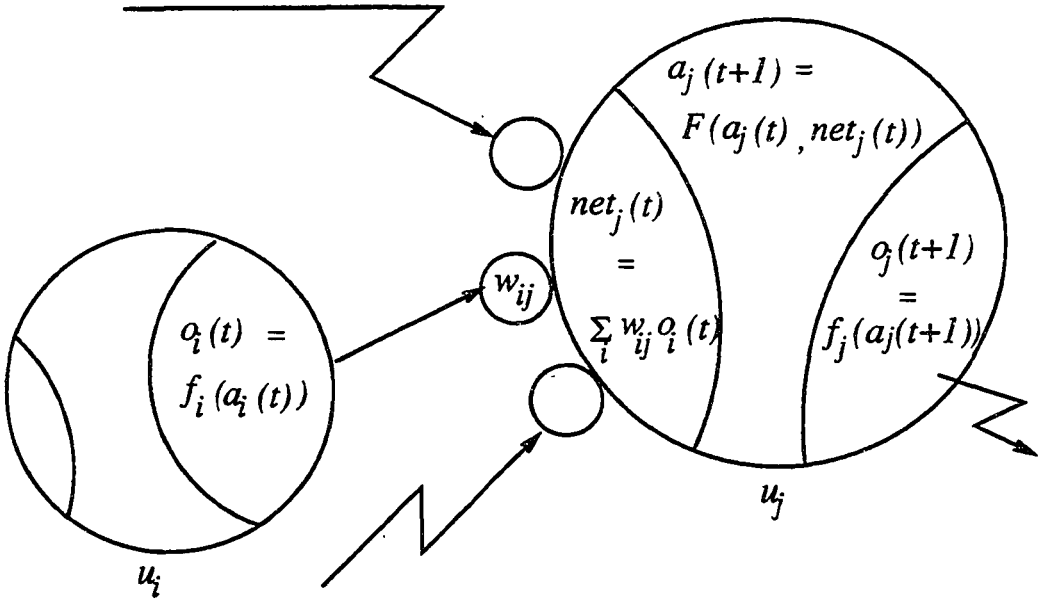

Figure 2.1: The processing of the j^{th} neuron at time $t+1$

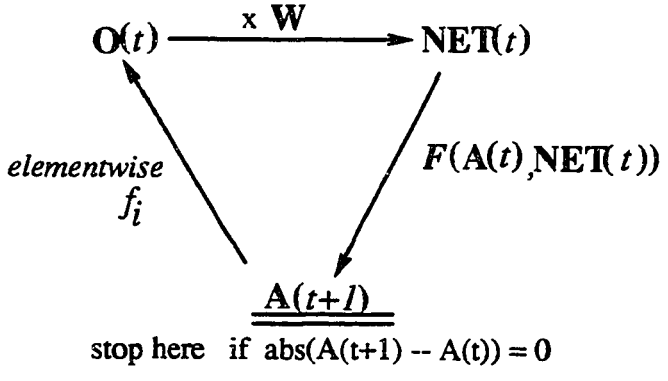

Figure 2.2: The operation cycle of a neural network

The operation of a neural network is illustrated using the example of XOR problem in Figure 2.3. The idea of the XOR problem is to have a system which responds 1 if it receives (0, 1) or (1, 0) as an input and responds 0 otherwise. Figure 2.3 shows an operation of a neural network using the data (1, 0).

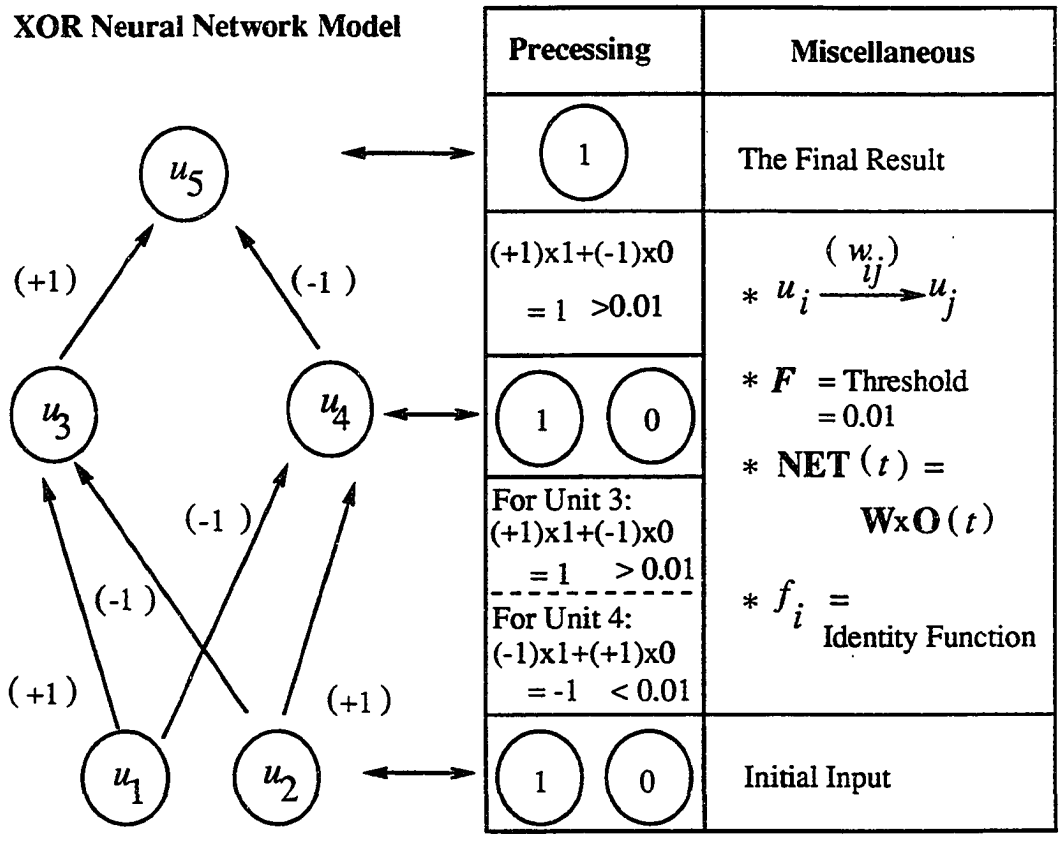

Figure 2.3: A neural network model for the XOR problem

As can be seen from Figure 2.3, a neural computing device relies upon massive parallelism and a high degree of connectivity among simple analog processors, which represents a radical departure from the conventional computer architecture. Unlike computations in the conventional computer systems, neural computations have no

notion of step-by-step sequencing. Instead, computation results from the collective emergent behavior of a dynamic system of simple processing elements.

2.3 Current Research Activities

This section provides a discussion of different aspects of the current research activities in neural networks.

2.3.1 Activation function

Several researchers have tried to find new activation functions to provide stronger recognization power to a neural network. Currently, *a linear function*, *a linear threshold function*, and *a sigmoid function* are used as representative activation functions. A linear function achieves no advantages from a multi-layer structure because of linearity. A linear threshold function is not differentiable and can't be used for generalized *delta learning rule* based on gradient descent. Hence, a sigmoid function which is a differentiable and non-linear function is commonly used. Williams' research work in [76] is an excellent guide for the analysis of different activation functions.

2.3.2 Learning

Their ability to learn is one of the most interesting characteristic of neural networks. Generally, learning means that a neural network is trained so that application of a set of input patterns produces the desired or consistent set of output patterns. With sequential application of input patterns, training is accomplished by adjusting the weight matrix $\mathbf{W}$ of a neural network according to a predetermined procedure.

Based on the procedure, learning procedures are classified into the following 2 different types.

1. **Supervised Learning:** In this learning procedure, training pairs of input and target patterns are required. After an input pattern is applied, an output pattern is derived from the network operation defined in Section 2.2. Then the difference (error) between the output pattern and the corresponding target pattern is fed back through the network to adjust the weight matrix **W** to minimize the error. The training should be done over all training pairs. Perceptron learning [13, 64, 79, 93], Backpropagation learning [64, 76, 79, 93], Adaline & Madaline learning [13, 79, 93], and Boltzmann learning [4, 23, 40, 79, 93] are learning procedures of this type.

2. **Unsupervised Learning:** In spite of success in many applications, supervised learning is criticized because sometimes it is very difficult to find target patterns. Hence, as a more biologically plausible model, unsupervised learning has been studied. The training set consists solely of input patterns without target patterns. This training procedure modifies the weight matrix **W** to produce output patterns that are consistent with input patterns on the view of similarity. In other words, the training procedure extracts the statistical properties of input patterns and groups similar patterns into classes. Hebbian learning, Kohonen's self-organization map, and Hecht-Nielsen's counter- propagation learning are the learning procedures of this type [57, 64, 79, 93].

The followings are current research issues for learning:

- Required size of the set of training patterns for given criteria of performance,

- Learning time, i.e., time to reach convergent values for weights,
- Relationship between learning time and internal data representation,
- The degree of generalization in classification, and
- The effect of learning new patterns on already *learned* patterns.

2.3.3 Mathematical analysis: capability and complexity

After learning procedures for multi-layer neural networks were introduced around 1980, many neural networks have shown promising experimental results for several applications. Many theoretical studies have been undertaken to explain the experimental results. The following issues are commonly studied.

1. **Capability:** In neural networks, a memory is defined as a stable state, i.e., an activation state remaining unchanged with network iterations as described in Figure 2.2. Hence, unlike the conventional memory in computer systems, deriving an explicit quantitative measure of memory capacity, based on dynamics of activation patterns of neural networks, is very difficult. Some researchers [55, 61] have applied qualitative approaches to analysis of the information capacity of a neural network. The other researchers [3, 6, 7, 43, 69, 95, 96] have used statistical neurodynamic methods to quantify the information capacity. In Chapter 3, theoretical studies of information capacity and the limitations of current approaches are rigorously reviewed. Besides, a more general statistical expression for the information capacity of Hopfield neural network is derived by replacing the 2 basic assumptions used in current quantitative approaches

by using Brown's Martingale Central Limit Theorem and the theory of *multivariate normal distribution.*

2. **Complexity of Convergence:** The number of iterations for a neural network to converge to a stable state is an issue. Abu-Mostafa [1], Fogelman [26, 27], and Goles [29, 30, 32] have shown that a neural network based on threshold activation function converges to a stable state within a polynomial time with respect to the number of neurons. But for real applications, their results should be reviewed under the consideration of the information capacity of neural networks. Sometimes the stable state may not be a stored state, which is a well-known problem called *spurious state problem* in the study of information capacity.

3. **Complexity of Learning:** Learning ability is an important characteristic of neural networks. Hence, answering the question, *How does learning time change with the problem size?*, is an important topic in study of neural networks. Judd formalized a notion of learning in a neural network characterizing the supervised training of feed-forward networks and proved that the general learning problem is **NP**-complete in [53] and [54]. Unfortunately, **NP**-completeness of learning means that there is no efficient general algorithm for learning. But he also argued that the difficulty of learning may be manageable through the following methods.

   - Network Modularization.
   - Probabilistic Approach.
   - Structurally Restricted Modeling of a Neural Network.

However, this field of neural network studies is still in primitive stage.

2.3.4 Performance analysis

The criteria for the performance analysis are somewhat subjective. For example, if we think of the information capacity as the ratio of the recallable patterns to stored patterns, *capacity* can also be classified as a criterion of performance. We restrict to performance analysis within 2 categories: *the usage of neurons; the usage of the connectivity of a neural network.* The research in this area is limited.

McClelland in [76] found that the distributions of activations of correct and spurious neurons pull apart as the number of learned pattern gets larger by using a probabilistic approach to a simple patterns associator model. Hence, his finding provides a base for the study of required number of neurons for a fixed number of learned patterns.

On the other hand, Abu-Mostafa in [2] proved that the entropy of a neural network becomes a lower bound for the connectivity of the network. To generalize their results, more rigorous study on the view of reliability is required.

2.3.5 Applications of neural networks

Applications of neural networks are so diversified that it is impossible to survey the corresponding research activities in detail. In general, current applications of neural networks are classified as follows:

- Combinatorial Optimization Problems,
- Frame-Like Concept Structure in the Field of AI,

- Pattern Recognition and Classification,
- Robotics Control Application,
- Signal Processing,
- Machine Vision, and
- Conceptualization of Modeling Complex Systems such as Economics, Biology, and Physics.

The best reference for the current state of the art in this area is the DARPA report [18], in which abstracts of contemporary applications of neural networks are well described according to the above categorization.

We provide some details on two application areas namely *combinatorial optimization problems* and *conceptual structures in AI.* Several well-known **combinatorial optimization problems** are **NP-complete** problems for which there is no general efficient algorithm. Therefore, the characteristic of polynomial time convergence with respect to the number of neurons in a neural network naturally appeals to computer scientists.

In 1985, Hopfield [46] provided some experimental results of using a neural network for the *Traveling Salesman Problem* (TSP). After then, many combinatorial optimization problems have been attacked with limited success, for example,

- Min Cut Problem in [12],
- Map and Graph Coloring in [16],
- Concentrator Assignment Problem in [71],

- Graph K-Partition, Vertex Cover, Maximum Independent Set, and Maximum Clique Problems in [73].

Generally, most researchers use the following approach to attack combinatorial optimization problems using a neural network.

Step 1 : From the combinatorial optimization problems,

1. Set an object function to be minimized, and
2. Set constraints for feasible solutions.

Step 2 : In a neural network,

1. Select a representation scheme in which a stable state is decoded into a solution to the optimization problem,
2. Map the object function and the constraints into a function to be minimized,
3. Transform the function to be minimized into a standard energy function, let's say, **E**,
4. Get the value of the weight matrix **W** from the energy function **E**,
5. Run the neural network which is stablized at a local optima of the energy function **E**, and
6. Interpret the solution to the problem from the stable state.

Neural network approach for solving combinatorial optimization problems was criticized by Wilson and Pawley [97] because of a problem in selecting some scaling parameters independent of a neural network structure. In fact, in some cases the

neural network does not guarantee even a feasible solution. Several researchers [19, 20, 37, 90] tried to solve the problem using some simulation techniques. But a standard way for selecting the scaling parameters remains an open problem.

Conceptual structures in AI is also an important research area. A major problem for AI scientists is to find a flexible structure for handling concepts. Through the last 20 years' study, it is a well-known fact that symbolic approach has limitations in generating a new concept from given concepts. Hence, neural networks having a role of associative memory based on dynamical activation of neurons attract AI scientists' attention. In Chapter 4, an implementation idea for a logical database system using neural network techniques is discussed along a sample system design.

2.3.6 Implementation technology

The implementation of neural networks can be classified into the following 3 classes [18, 79, 91].

1. **Software Implementation:** Software implementation that is created on a machine that is not made explicitly for artificial neural networks.
    - Super Computers.
    - Massively Parallel Computers.
    - Conventional Computers.
2. **Electonic Implementation:** Electronic implementation that is made with the sole purpose of performing artificial neural network processing.
    - Bus-Oriented Processors.

- Coprocessors.
- Integrated Circuits.

3. **Optical/Electro-Optical Implementations:** Any implementation in which optical components are used.

3. PROBABILISTIC INFORMATION CAPACITY OF HOPFIELD ASSOCIATIVE MEMORY

3.1 Introduction

The ability of recalling memorized patterns is a very important feature of the human memory. In 1982, Hopfield [43] rekindled the interest in networks of automata by introducing a new kind of associative memory based on a simple neural network model. Computational properties of his network model emerge as collective properties of a system having a large number of simple neurons. Establishing empirically that such collective properties include default assignment, error correction, and spontaneous generalization, he demonstrated the attractiveness of his model for many applications.

An important next step in understanding the Hopfield network model is to mathematically quantify its performance as a memory. In the Hopfield model, a memory is defined as a stable state, i.e., an activation state which remains unchanged with network iterations. In other words, the information capacity of the Hopfield model is based on the dynamics of activation patterns in the neural network. In contrast to the standard memory model, where the information capacity is an explicit quantity based on the number of memory bits, estimating the information capacity of the Hopfield associative memory is considerably more complex.

The number of patterns that can be stored in a Hopfield network depends on the given set of patterns. For example, it may be possible to store a specific set of m patterns as stable states but that does not imply that any set of m patterns can be stored as stable states assuming the same size of the network. This dependence of information capacity on the set of patterns has prompted researchers [6, 7, 43, 69, 95, 96] to consider the storage of random patterns and to use probabilistic estimates of the information capacity of the Hopfield network. We formalize the notion of probabilistic information capacity of the Hopfield network. This notion is intrinsic in earlier studies but they do not state it explicitly and precisely. The formalization helps to clarify the previous work [6, 7, 43, 69, 95, 96] and more importantly, it paves the way for introducing powerful statistical techniques to analyze the information capacity.

Some researchers [55, 61] have tried to study the information capacity with qualitative analysis, however their approaches have not led to any concrete results. Our study is quantitative; we provide numerical estimates of information capacity based on the probability of success in storing random binary patterns. Amari [6, 7] and McEliece [69] pioneered the use of statistical techniques to study the capacity of Hopfield network. On the other hand, Weisbuch's [96] statistical techniques use additional assumptions to provide results of information capacity more consistent with simulation than the other studies [6, 7, 43, 69]. We show that the capacity heuristics can be rendered more accurate and some inexact assumptions, made in earlier studies for the purpose of simplifying the analysis, can be eliminated with the help of more powerful statistical techniques. We establish the connection between the dynamics of the Hopfield network and the theory of multivariate normal distribution [9, 11, 34].

Using Brown's Martingale Central Limit Theorem [11] and Gupta's transformation [34], we derive a mathematical expression for probabilistic information capacity of the Hopfield memory. The numerical results can be derived for networks of large size from the mathematical expression by using a standard statistical software package.

The accuracy of our method is established by comparing it with known simulation results. Simulation studies of Hopfield network take a lot of computation time if the network size is large. Hopfield did a simulation study of information capacity based on networks up to the size of 100 neurons. We compare our results with the results of Weisbuch's experimental study of up to 500-neuron network.

In Section 3.2, the Hopfield's model of associative memory is reviewed and the notion of the probabilistic information capacity is formalized. Section 3.3 examines the current approaches to the capacity heuristics and we propose changes to the current approaches based on statistical neurodynamics. In Section 3.4, our approach for analyzing the capacity of the Hopfield memory is discussed along with main theorems. In Section 3.5, the performance results based on our mathematical analysis are compared with the results from other theoretical [6, 7, 69, 95] and experimental [96] studies. Section 3.6 provides the concluding remarks.

3.2 The Hopfield Associative Memory

This section provides a self-contained description of the Hopfield Associative Memory and introduces the precise mathematical notion of probabilistic information capacity.

The Hopfield neural network model of associative memory consists of n pairwise connected neurons. Any neuron i can be in one of two states: $v_i = 0$ (off) or $v_i = 1$

(on).

Definition 3.1 : A state vector $\mathbf{V} = [v_1, \ldots, v_n]$, is defined to be a binary vector whose i^{th} component corresponds to the state of the i^{th} neuron.

Definition 3.2 : A connection matrix is the $n \times n$ matrix $\mathbf{W} = (w_{i,j})$, where the $(i,j)^{th}$ entry of $\mathbf{W}$ is the strength of synaptic connection from neuron i to neuron j.

Each choice of $\mathbf{W}$ defines a specific neural network of n neurons. In other words, the collective behavior of the neural network is entirely specified by $\mathbf{W}$. In fact, the matrix $\mathbf{W}$ acts as a decoding machine which can be recognized as a kind of information storage. Hopfield model requires that $w_{i,j} = w_{j,i}$ and $w_{i,i} = 0$.

According to the Hebbian learning rule [69], to memorize (store) m patterns (state vectors) $\mathbf{V}^1, \mathbf{V}^2, \ldots, \mathbf{V}^m$ in the Hopfield neural network, each entry of the connection matrix $\mathbf{W}$ is computed by

$$w_{i,j} = \sum_{s=1}^{m} (2v_i^s - 1) \cdot (2v_j^s - 1) \tag{3.1}$$

which can be simplified with

$$w_{i,j} = \sum_{s=1}^{m} X_i^s \cdot X_j^s \tag{3.2}$$

where X_i^s is $(2v_i^s - 1)$.

A randomly selected neuron receives inputs from connected neurons and changes its state in the following manner at each discrete time step t :

$$v_i(t) = sgn\left\{ \sum_{j=1}^{n} w_{i,j} \cdot v_j(t-1) \right\} = \mathbf{T}(\mathbf{V}(t-1)) \tag{3.3}$$

where $sgn(y) = \begin{cases} 1 & \text{if } y \geq 0 \\ 0 & \text{otherwise} \end{cases}$ and $v_i(t)$ represents the state of the i^{th} neuron at time t. $\mathbf{T}$ is a non-linear state transition operator.

Finally, the recalling process of the memorized (stored) patterns can be described as follows: Start with an initial state represented by a binary vector $\mathbf{V}(0)$. The state is changed iteratively according to Equation (3.3). The iterative process is repeated until a state that remains unchanged with further network iterations is reached. The resulting state, let's say $\mathbf{V}^1$, $\mathbf{V}^1$ is said to be *recalled* from $\mathbf{V}(0)$.

Definition 3.3 : A state vector (a pattern) $\mathbf{V}$ is called a *stable state* iff $\mathbf{V}$ is recalled from $\mathbf{V}$, i.e., iff from Equation (3.3)

$$\forall i \in \{1, \ldots, n\}, \quad v_i \cdot (\sum_{j=1}^{n} w_{i,j} v_j) \geq 0 \tag{3.4}$$

Let $\mathbf{S}_m$ denote a set of m random binary patterns $\mathbf{V}^1$, ..., $\mathbf{V}^m$ each size of n. Consider a Hopfield network where the weights are as given by the Hebbian learning (Equation 3.1) to store all the patterns in $\mathbf{S}_m$. Let $\mathbf{P}(\mathbf{S}_m)$ denote the probability that all the patterns in $\mathbf{S}_m$ are in fact stable patterns.

Definition 3.4 : Given an $\alpha \in [0, 1]$ *the information capacity* $\mathbf{I}_\alpha$ is defined to be the maximum integer m such that $\mathbf{P}(\mathbf{S}_m) \geq \alpha$.

In this thesis, we study the information capacity of the Hopfield memory based on statistical neurodynamics characterizing the non-linear state transition operator $\mathbf{T}$, in which $\mathbf{W}$ is a random connection matrix.

3.3 Information Capacity Heuristics

In this section, we first review the current approaches for estimating the information capacity of the Hopfield Associative Memory. The quantitative approaches are reviewed in more depth to set the stage for our work. After the review, we discuss the main idea behind our extension of the current approaches. This section uses the framework developed in Section3.2 along with the following definitions:

Definition 3.5 : Let d be a positive integer, a pattern $\mathbf{V}$ is called a *d–attractor* iff $\forall\, \mathbf{V}'$ whose Hamming distance from $\mathbf{V}$ is less then or equal to d, $\mathbf{V}$ is recalled from $\mathbf{V}'$.

Definition 3.6 : The *basin of attraction* of a stored pattern $\mathbf{V}$ is defined as the locus of all vectors in the state space which are attracted to $\mathbf{V}$.

3.3.1 Qualitative heuristics

In 1986, Keeler [55] tried to analyze the capacity of the Hopfield memory by investigating the basin of attraction as a function of the number of stored patterns in the network. The basins of attraction were explored by taking a random two dimensional rectangle through the state space. Keeler defined the two dimensional rectangle as follows: Given a stored pattern $\mathbf{A}$, ramdomly select a pattern $\mathbf{B}$ whose Hamming distance is d (he used $\lfloor n/2 \rfloor$ for d which corresponds to maximum possible Hamming distance). Then, construct a rectangle using $\mathbf{A}$, $\bar{\mathbf{A}}$, $\mathbf{B}$, $\bar{\mathbf{B}}$, where $\bar{\mathbf{A}}$ and $\bar{\mathbf{B}}$ correspond to the complements of $\mathbf{A}$ and $\mathbf{B}$ respectively.

A lattice point, say (i,j), in the rectangle denotes a pattern $\mathbf{C}$ which differs from the stored pattern $\mathbf{A}$ by i Hamming units in the d neurons at which $\mathbf{A}$ and $\mathbf{B}$

differ and by j Hamming units in the remaining $(n - d)$ neurons. For each lattice point in the rectangle, Equation (3.3) is applied until it reaches a stable state. If the associated stable state is the stored pattern **A**, then the point is labeled with a black dot. If not, it is left blank. Thus the two dimensional rectangle is divided into two different regions: the black region and the blank region. The black region corresponds to the basin of the attractor **A**.

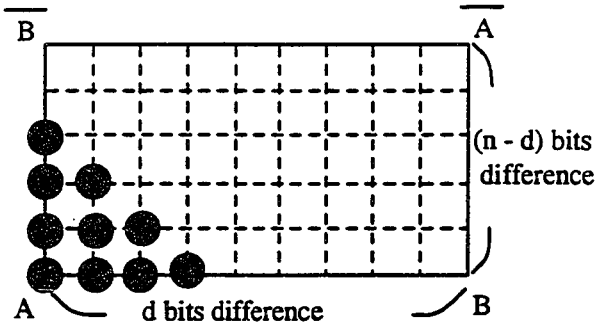

Figure 3.1: An example of the basin for an attractor A

Based on some examples, Keeler found that as the number of stored patterns increases, the degradation of the basin occurs around $m = 0.15n$ which is Hopfield's simulation result for information capacity [43]. But some way of quantifying Keeler's result remained a problem to be solved.

Krowitz [61] graphically described the activation state of neurons in a neural network using Equation (3.3) and provided some graphic tools. Unfortunately, he also couldn't quantify the result of his qualitative analysis.

3.3.2 Current quantitative heuristics and proposed extension

Several researchers (Amari [6, 7], Hopfield [43], McEliece [69], Weisbuch [95, 96], etc.) have proposed statistical neurodynamical methods to explore the capacity of the Hopfield memory. In this section, their studies are reviewed. The main idea of our research, which extends their works, is also briefly discussed.

Most statistical methods depend upon two invariance conditions derived from Equation (3.2), Equation (3.3) and Equation (3.4). Let the number of neurons be n and the number of stored patterns be m. The invariance conditions (ICs) are :

IC 3.1 : Given a stored pattern $\mathbf{V}^s$, an off-state neuron i in the pattern remains *off* (0) if

$$0 > \sum_{j=1,j\neq i}^{n} w_{i,j} \cdot v_j^s = -\sum_{j=1,j\neq i}^{n} v_j^s + \sum_{s'=1,s'\neq s}^{m} \sum_{j=1,j\neq i}^{n} X_i^{s'} \cdot X_j^{s'} \cdot v_j^s \quad (3.5)$$

IC 3.2 : Given a stored pattern $\mathbf{V}^s$, an on-state neuron i in the pattern remains *on* (1) if

$$0 \leq \sum_{j=1,j\neq i}^{n} w_{i,j} \cdot v_j^s = \sum_{j=1,j\neq i}^{n} v_j^s + \sum_{s'=1,s'\neq s}^{m} \sum_{j=1,j\neq i}^{n} X_i^{s'} \cdot X_j^{s'} \cdot v_j^s \quad (3.6)$$

To use the invariance conditions for statistical analysis of the information capacity, all of us (Amari, McEliece, Weisbuch, and we) require the following assumption.

Assumption 3.1 : For s = 1, ..., m and i = 1, ..., n, all v_i^s are identically, independently distributed (*i.i.d.*) random variables taking 1 and 0 with probability 0.5.

For given s and i, define the signal term (S_i^s) as $\sum_{j=1,j\neq i}^{n} v_j^s$ and the noise term (N_i^s) as $\sum_{s'=1,s'\neq s}^{m} \sum_{j=1,j\neq i}^{n} X_i^{s'} \cdot X_j^{s'} \cdot v_j^s$. We want to evaluate the probability

that the invariance condition for a neuron is satisfied. Because of symmetry, the probability is the same for both the invariance conditions. The following discussion is based on IC 3.1.

The existing studies due to Amari, McEliece, and Weisbuch require an additional assumption.

Assumption 3.2 : Each element in the noise term, $X_i^{s'} \cdot X_j^{s'} \cdot v_j^s$ where $s' \neq s$, $s' = 1, \ldots, m$, $j \neq i$ and $j = 1, \ldots, n$ is an *i.i.d.* random variable.

Based on these assumptions (3.1 and 3.2) and the Central Limit Theorem, the current capacity heuristics lead to the result:

$$-S_i^s + N_i^s \sim \mathbf{N}\left(\frac{-(n-1)}{2}, \frac{(m-1)(n-1)}{2}\right) \tag{3.7}$$

where **N** means a normal distribution.

Assumption 3.2 is introduced to simplify the analysis and it is not accurate because the elements of the noise terms are in fact not independent of each other. In Theorem 3.1, we prove the result without using Assumption 3.2:

$$-S_i^s + N_i^s \sim \mathbf{N}\left(\frac{-(n-1)}{2}, \frac{(2m-1)(n-1)}{4}\right) \tag{3.8}$$

for all $s = 1, \ldots, m$, and $i = 1, \ldots, n$.

The existing studies derive the following result from Equation (3.7): The probability that a neuron satisfies the invariance condition is given by

$$\mathbf{P}(S_i^s > N_i^s) = \Phi(z) = \int_{-\infty}^{z} \phi(t)dt \tag{3.9}$$

where $\phi(t) = (1/\sqrt{2\pi})e^{-t^2/2}$ for $-\infty < t < \infty$ and $z = \sqrt{(n-1)}/\sqrt{2(m-1)}$.

Remark : In our case, by Equation (3.8), $\mathbf{P}(S_i^s > N_i^s) = \Phi(z')$ where $z' = \sqrt{(n-1)}/\sqrt{(2m-1)}$.

From Equation (3.9), McEliece [69] approximated the expected number of failed neurons (not satisfying the invariance condition) in a stored pattern to $n\,(1-\Phi(z))$. Making the conjecture that the number of failed neurons approximately follows Poisson distribution, he derived the probability that a stored pattern is indeed a stable state (fixed point) as follows: For a fixed probability β (for example, $\beta = 0.9999$),

$$\beta = \exp\{-n(1-\Phi(z))\} = \exp\left\{-n\left(1-\Phi\left(\frac{\sqrt{(n-1)}}{\sqrt{2(m-1)}}\right)\right)\right\} \tag{3.10}$$

Remark : If $X \sim Poisson(\lambda)$ for a fixed λ, then $\mathrm{P}(X=x) = (e^{-\lambda}\cdot\lambda^{x})/x!$. In particular $\mathrm{P}(X=0) = \mathrm{P}(\text{no failure}) = e^{-\lambda}$.

From Equation (3.10), McEliece derives the approximation that $m \sim \frac{n}{2\log n}$ with the assumption that $n\,(1-\Phi(z))$ is a constant for all n, which is not necessarily the case. On the other hand, Amari [6, 7] and Weisbuch [95, 96] used the following additional assumption:

Assumption 3.3 : $\forall s = 1,\ldots,m$, $\forall i = 1,\ldots,n$, each term $-S_i^s + N_i^s$ which is a random variable defining the invariance condition for a neuron is independent of each other.

Using Equation (3.9) and the Assumption 3.3, they concluded that the probability that all stored patterns are stable is

$$\mathbf{P} = (\mathbf{P}(S_i^s > N_i^s))^{nm} = \{\Phi(z)\}^{nm} \tag{3.11}$$

Amari [6] acknowledges that the dependency among the random variables $-S_i^s + N_i^s$ for $1 \le i \le n$ and $1 \le s \le m$, can not be neglected through some simulations. In our proposed heuristic, Assumption 3.3 is used, instead we account

for the dependency among the random variables used in the assumption (Theorem 3.2 and Theorem 3.3).

In summary, with the help of more powerful techniques from statistics, we will eliminate the use of incorrect assumptions 3.2 and 3.3 which so far were used to simplify the analysis. In Theorem 3.4, we derive a statistical expression for the information capacity by using the result from Theorem 3.2, Theorem 3.3 and Gupta's transformation technique [34].

3.4 Capacity Based on Multivariate Normal Approximation

In this section, we justify our claims in Section 3.2. by providing rigorous proofs. Our proofs are based on the invariance condition given by Equation (3.5) and Assumption 3.1. Let

$$\mathbf{A}(i,s) = -S_i^s + N_i^s = -\sum_{j=1,j\neq i}^{n} v_j^s + \sum_{s'=1,s'\neq s}^{m} \sum_{j=1,j\neq i}^{n} X_i^{s'} \cdot X_j^{s'} \cdot v_j^s \quad (3.12)$$

Proposition 3.1 : Under the Assumption 3.1, approximately,

$$S_i^s \sim \mathbf{N}\left(\frac{(n-1)}{2}, \frac{(n-1)}{4}\right)$$

Proof: By the Assumption 3.1, $S_i^s \sim \mathbf{B}\left((n-1), \frac{1}{2}\right)$ where $\mathbf{B}(n, p)$ is a binomial distribution with parameters n and p. Then by the Central Limit Theorem, we can approximate $\mathbf{B}\left((n-1), \frac{1}{2}\right)$ by $\mathbf{N}\left(\frac{(n-1)}{2}, \frac{(n-1)}{4}\right)$ for large n. □

Proposition 3.2 : Under the Assumption 3.1, the elements $X_i^{s'} \cdot X_j^{s'} \cdot v_j^s$ in N_i^s for $1 \leq i,j \leq n$, $i \neq j$, $1 \leq s, s' \leq m$, and $s' \neq s$, are mutually uncorrelated.

Proof: Fix i & s and select two different elements $X_i^{s'1} \cdot X_{j_1}^{s'1} \cdot v_{j_1}^s$ and $X_i^{s'2} \cdot X_{j_2}^{s'2} \cdot v_{j_2}^s$ such that $i \neq j_1$, $i \neq j_2$, $s \neq s'_1$, and $s \neq s'_2$. By Assumption 3.1 and

Equation (3.2),

$$\mathbf{E}(X_i^s) = 0, \ \forall\, s \ \ 1 \le s \le m. \tag{3.13}$$

Therefore,

$$\mathbf{E}(X_i^{s'1} \cdot X_{j_1}^{s'1} \cdot v_{j_1}^s) = \mathbf{E}(X_i^{s'1}) \cdot \mathbf{E}(X_{j_1}^{s'1} \cdot v_{j_1}^s) = 0. \tag{3.14}$$

Hence, by Equation (3.14) and Assumption 3.1,

$$\begin{aligned}
&\mathbf{Cov}(X_i^{s'1} \cdot X_{j_1}^{s'1} \cdot v_{j_1}^s,\ X_i^{s'2} \cdot X_{j_2}^{s'2} \cdot v_{j_2}^s) \\
&= \mathbf{E}(X_i^{s'1} \cdot X_{j_1}^{s'1} \cdot v_{j_1}^s \cdot X_i^{s'2} \cdot X_{j_2}^{s'2} \cdot v_{j_2}^s) \\
&= \mathbf{E}(X_i^{s'1} \cdot X_i^{s'2}) \cdot \mathbf{E}(X_{j_1}^{s'1} \cdot X_{j_2}^{s'2}) \cdot \mathbf{E}(v_{j_1}^s \cdot v_{j_2}^s) \\
&= 0. \quad \square
\end{aligned}$$

Remark : While these random variables $(X_i^{s'} \cdot X_j^{s'} \cdot v_j^s)$ are identically distributed and are mutually uncorrelated, they can not be independent of each other.

Proposition 3.3 : Under the Assumption 3.1,
$\mathbf{E}(N_i^s) = 0$ and $\mathbf{Var}(N_i^s) = \frac{(m-1)(n-1)}{2}$ for $1 \le i \le n$ and $1 \le s \le m$.

Proof: Fix i & s and select two different elements $X_i^{s'1} \cdot X_{j_1}^{s'1} \cdot v_{j_1}^s$ and $X_i^{s'2} \cdot X_{j_2}^{s'2} \cdot v_{j_2}^s$.
By Equation (3.14),

$$\mathbf{E}(N_i^s) = \sum_{s'=1, s' \ne s}^{m} \ \sum_{j=1, j \ne i}^{n} \mathbf{E}(X_i^{s'} \cdot X_j^{s'} \cdot v_j^s) = 0.$$

And, by Proposition 3.2,

$$\mathbf{Var}(N_i^s) = \sum_{s'=1,s'\neq s}^{m} \sum_{j=1,j\neq i}^{n} \mathbf{Var}(X_i^{s'} \cdot X_j^{s'} \cdot v_j^s) \; +$$

$$\sum_{s_1'=1,s_1'\neq s}^{m} \sum_{s_2'=1,s_2'\neq s}^{m} \sum_{j_1=1,j_1\neq i}^{n} \sum_{j_2=1,j_2\neq i}^{n} \mathbf{Cov}(X_i^{s_1'} \cdot X_{j_1}^{s_1'} \cdot v_{j_1}^s,\; X_i^{s_2'} \cdot X_{j_2}^{s_2'} \cdot v_{j_2}^s)$$

$$= \sum_{s'=1,s'\neq s}^{m} \sum_{j=1,j\neq i}^{n} \mathbf{Var}(X_i^{s'} \cdot X_j^{s'} \cdot v_j^s),$$

because the set of random variables

$\{X_i^{s'} \cdot X_j^{s'} \cdot v_j^s,\ 1 \le i,j \le n,\ 1 \le s,s' \le m\}$ are identically distributed,

$$= (m-1)(n-1)\mathbf{Var}(X_i^1 \cdot X_1^1 \cdot v_1^s)$$

$$= (m-1)(n-1)\{\mathbf{E}((X_i^1 \cdot X_1^1)^2 \cdot (v_1^s)^2) - \mathbf{E}(X_i^1) \cdot \mathbf{E}(X_1^1 \cdot v_1^s)\}$$

$$= (m-1)(n-1)\mathbf{E}(v_1^s)^2$$

$$= \frac{(m-1)(n-1)}{2}. \quad \square$$

Proposition 3.4 : Under the Assumption 3.1, approximately

$$N_i^s \;\sim\; \mathbf{N}\left(0,\, \frac{(m-1)(n-1)}{2}\right)$$

Proof : By Proposition 2.2, Proposition 3.3, and Brown's Martingale Central Limiting Theorem (refer to [11] and Appendix A). $\square$

Proposition 3.5 : S_i^s and N_i^s are uncorrelated.

Proof:

$$\mathbf{Cov}(S_i^s,\, N_i^s) \;=\; \mathbf{Cov}\Big(\sum_{j=1,j\neq i}^{n} v_j^s,\; \sum_{s'=1,s'\neq s}^{m} \sum_{j=1,j\neq i}^{n} X_i^{s'} \cdot X_j^{s'} \cdot v_j^s\Big)$$

$$= \sum_{j_1=1, j_1\neq i}^{n} \sum_{j_2=1, j_2\neq i}^{n} \sum_{s'=1, s'\neq s}^{m} \mathbf{Cov}(v_{j_1}^s,\ X_i^{s'} \cdot X_{j_2}^{s'} \cdot v_{j_2}^s)$$
$$= 0.$$

Because, by Equation (3.13), $\forall\, j_1,\ j_2$ and s',

$$\begin{aligned}\mathbf{Cov}(v_{j_1}^s,\ X_i^{s'} \cdot X_{j_2}^{s'} \cdot v_{j_2}^s) &= \mathbf{E}(v_{j_1}^s \cdot X_i^{s'} \cdot X_{j_2}^{s'} \cdot v_{j_2}^s) \\ &= \mathbf{E}(v_{j_1}^s)\mathbf{E}(X_i^{s'})\mathbf{E}(X_{j_2}^{s'} \cdot v_{j_2}^s) \\ &= 0 \quad \square\end{aligned}$$

Proposition 3.6 : If $S_1 \xrightarrow{d} N_1$ and $S_2 \xrightarrow{d} N_2$ where N_1 and N_2 are uncorrelated normal random variables, and d denotes convergence in distribution, then $S_1 + S_2 \xrightarrow{d} N_1 + N_2$.

Proof: A well-known statistical theorem (refer to [10]). $\square$

Theorem 3.1 : Under Assumption 3.1, approximately

$$-S_i^s + N_i^s \sim \mathbf{N}\left(\frac{-(n-1)}{2},\ \frac{(n-1)(2m-1)}{4}\right).$$

Proof: By Proposition 3.1 $\left(-S_i^s \sim \mathbf{N}\left(\frac{-(n-1)}{2},\ \frac{(n-1)}{4}\right)\right)$, Proposition 3.4 $\left(N_i^s \sim \mathbf{N}\left(0,\ \frac{(n-1)(m-1)}{2}\right)\right)$, Proposition 3.5, and Proposition 3.6. $\square$

Remark : With the assumption that n is fixed and m is large, McEliece claimed that $-S_i^s + N_i^s \sim \mathbf{N}\left(\frac{-(n-1)}{2},\ \frac{(n-1)(m-1)}{2}\right)$. In other words, the effect of the variance of S_i^s $\left(i.e.,\ \frac{(n-1)}{4} \ll \frac{(n-1)(m-1)}{2}\right)$ is neglected.

Corollary 3.1 : The probability that a neuron satisfies the invariance condition is

$$\mathbf{P}(-S_i^s + N_i^s < 0) = \Phi(\sqrt{(n-1)}/\sqrt{(2m-1)}) =$$
$$\int_{-\infty}^{\sqrt{(n-1)}/\sqrt{2m-1}} (1/\sqrt{2\pi}) \cdot e^{-t^2/2}\, dt \quad \text{for } -\infty < t < \infty.$$

Now we need to answer the questions what is the probability that n neurons in a stored pattern satisfy the invariance condition, more generally, what is the probability that nm neurons of m stored patterns satisfy the invariance condition. To answer these questions, first we check the dependency among the terms $-S_i^s + N_i^s$ for n neurons in a stored pattern. Next, we check the dependency among the terms $-S_i^s + N_i^s$ for i^{th} $(1 \leq i \leq n)$ neurons in m stored patterns.

Proposition 3.7 : $\mathbf{Cov}(\mathbf{A}(1,s), \mathbf{A}(2,s)) = \frac{(n+m-3)}{4}$, $\forall\, s\ 1 \leq s \leq m$.

Proof: Note that

$$\begin{aligned}
\mathbf{Cov}(S_1^s, S_2^s) &= \mathbf{Cov}\Big(\sum_{j\neq 1, j=2}^{n} v_j^s, \sum_{j=1, j\neq 2}^{n} v_j^s\Big) \\
&= \sum_{j_1\neq 1, j_1=2}^{n} \sum_{j_2=1, j_2\neq 2}^{n} \mathbf{Cov}(v_{j_1}^s, v_{j_2}^s) \\
&= \frac{(n-2)}{4},
\end{aligned}$$

and by the same arguments used in Proposition 3.5,

$$\mathbf{Cov}(S_1^s, N_2^s) = \mathbf{Cov}(S_2^s, N_1^s) = 0,$$

also,

$$\begin{aligned}
&\mathbf{Cov}(N_1^s, N_2^s) \\
&= \mathbf{Cov}\Big(\sum_{s'=1, s'\neq s}^{m} \sum_{j=1, j\neq i}^{n} X_1^{s'} \cdot X_j^{s'} \cdot v_j^s, \sum_{s'=1, s'\neq s}^{m} \sum_{j=1, j\neq i}^{n} X_2^{s'} \cdot X_j^{s'} \cdot v_j^s\Big) \\
&= \\
&\sum_{s_1'=1, s_1'\neq s}^{m} \sum_{s_2'=1, s_2'\neq s}^{m} \sum_{j_1\neq 1, j_1=2}^{n} \sum_{j_2=1, j_2\neq 2}^{n} \mathbf{Cov}(X_1^{s_1'} \cdot X_{j_1}^{s_1'} \cdot v_{j_1}^s, X_2^{s_2'} \cdot X_{j_2}^{s_2'} \cdot v_{j_2}^s),
\end{aligned}$$

because by Proposition 3.2, the terms of **Cov** will be 0 except in the cases $(s_1' = s_2' = s, j_1 = 1, j_2 = 2)$ and $(s_1' = s_2' = s, j_1 = 2, j_2 = 1)$,

$$\begin{aligned}
&= \sum_{s'=1, s' \neq s}^{m} \sum_{(j_1=2, j_2=1)} \mathbf{Cov}(X_1^{s'} \cdot X_{j_1}^{s'} \cdot v_{j_1}^{s}, X_2^{s'} \cdot X_{j_2}^{s'} \cdot v_{j_2}^{s}) \\
&= (m-1)\mathbf{E}(X_1^{s'} \cdot X_2^{s'} \cdot v_2^{s} \cdot X_2^{s'} \cdot X_1^{s'} \cdot v_1^{s}) \\
&= (m-1)\mathbf{E}(v_1^{s} \cdot v_2^{s}) \\
&= (m-1)\mathbf{E}(v_1^{s}) \cdot \mathbf{E}(v_2^{s}) \\
&= \frac{(m-1)}{4}.
\end{aligned}$$

Therefore,

$$\begin{aligned}
\mathbf{Cov}(\mathbf{A}(1,s), \mathbf{A}(2,s)) &= \frac{(n-2)}{4} + 0 + \frac{(m-1)}{4} \\
&= \frac{(n+m-3)}{4}. \quad \square
\end{aligned}$$

Theorem 3.2 : The correlation coefficient $\rho_{A(1,s),A(2,s)}$ of $\mathbf{A}(1,s)$ and $\mathbf{A}(2,s)$ is $\frac{(n+m-3)}{(n-1)(2m-1)}$ for $1 \leq s \leq m$.

Proof: By Theorem 3.1, $\sigma^2_{A(1,s)} = \sigma^2_{A(2,s)} = \frac{(n-1)(2m-1)}{4}$, and by Proposition 3.7, $\mathbf{Cov}(\mathbf{A}(1,s), \mathbf{A}(2,s)) = \frac{(n+m-3)}{4}$. Hence,

$$\begin{aligned}
\rho_{A(1,s),A(2,s)} &= \frac{\mathbf{Cov}(\mathbf{A}(1,s), \mathbf{A}(2,s))}{(\sigma_{A(1,s)} \cdot \sigma_{A(2,s)})} \\
&= \frac{(n+m-3)}{((n-1)(2m-1))}. \quad \square
\end{aligned}$$

Remark : $\lim_{n,m \to \infty} \rho_{A(1,s),A(2,s)} = 0$. Hence, for large n and m, we may be able to assume that there is no dependency among the terms $-S_i^s + N_i^s$ for n neurons within a stored pattern.

Proposition 3.8 : Under Assumption 3.1,

$$\mathbf{Cov}(\mathbf{A}(i,1),\mathbf{A}(i,2)) = (n-1)(m-2)/4 \text{ for } i \ \ 1 \le i \le n.$$

Proof: Note that by Assumption 3.1,

$$\begin{aligned}\mathbf{Cov}(S_i^1, S_i^2) &= \mathbf{Cov}(\sum_{j=1,j\neq i}^{n} v_j^1, \sum_{j=1,j\neq i}^{n} v_j^2) \\ &= \sum_{j_1=1,j_1\neq i}^{n}\sum_{j_2=1,j_2\neq i}^{n} \mathbf{Cov}(v_{j_1}^1, v_{j_2}^2) \\ &= 0,\end{aligned}$$

and by the same arguments used in Proposition 3.5,

$$\mathbf{Cov}(S_i^1, N_i^2) = \mathbf{Cov}(S_i^2, N_i^1) = 0,$$

also,

$$\begin{aligned}&\mathbf{Cov}(N_i^1, N_i^2) \\ &= \mathbf{Cov}(\sum_{s'\neq 1,s'=2}^{m}\sum_{j=1,j\neq i}^{n} X_i^{s'}\cdot X_j^{s'}\cdot v_j^1, \sum_{s'=1,s'\neq 2}^{m}\sum_{j=1,j\neq i}^{n} X_i^{s'}\cdot X_j^{s'}\cdot v_j^2) \\ &= \\ &\sum_{s_1'\neq 1,s_1'=2}^{m}\sum_{s_2'=1,s_2'\neq 2}^{m}\sum_{j_1=1,j_1\neq i}^{n}\sum_{j_2=1,j_2\neq i} \mathbf{Cov}(X_i^{s_1'}\cdot X_{j_1}^{s_1'}\cdot v_{j_1}^1, X_i^{s_2'}\cdot X_{j_2}^{s_2'}\cdot v_{j_2}^2),\end{aligned}$$

because the terms of **Cov** will be 0 except in the case

$(s_1' = s_2' = s,\ j_1 = j_2 = j)$ by Proposition 3.2,

$$\begin{aligned}&= \sum_{s'\neq 1,2,s'=3}^{m}\sum_{j=1,j\neq i}^{n} \mathbf{E}(X_i^{s'}\cdot X_j^{s'}\cdot v_j^1\cdot X_i^{s'}\cdot X_j^{s'}\cdot v_j^2) \\ &= \sum_{s'=3}^{m}\sum_{j=1,j\neq i}^{n} \mathbf{E}(v_j^1\cdot v_j^2) \\ &= \frac{(n-1)(m-2)}{4}.\end{aligned}$$

Therefore,

$$\mathbf{Cov}(\mathbf{A}(i,1),\mathbf{A}(i,2)) \;=\; \frac{(n-1)(m-2)}{4}. \quad \square$$

Theorem 3.3 : The correlation coefficient $\rho_{A(i,1),A(i,2)}$ of $\mathbf{A}(i,1)$ and $\mathbf{A}(i,2)$ is $\frac{(m-2)}{(2m-1)}$, for $1 \le i \le n$.

Proof: By the same arguments in the proof of Theorem 3.2. $\square$

Remark : $\lim_{m\to\infty} \rho_{A(i,1),A(i,2)} = \frac{1}{2}$. Hence, the dependency among the terms $-S_i^s + N_i^s$ for i^{th} neurons in m stored patterns can not be neglected (recall Assumption 3.3 and Amari's note).

Now, those dependencies both among the random variables (i.e., $-S_i^s + N_i^s$) for the invariance condition of n neurons in a stored pattern and among the random variables for the invariance condition of i^{th} neurons in m stored patterns should be considered to derive a statistical expression for the information capacity.

Because the former dependency is negligible by the Remark following Theorem 3.2, we derive a statistical expression considering only the latter dependency. From Theorem 3.1, Proposition 3.8, and Theorem 3.3, we can derive the following.

$$\begin{pmatrix} U_1 \\ \vdots \\ \vdots \\ U_m \end{pmatrix} \;\sim\; \mathbf{N}\left(\begin{pmatrix} 0 \\ \vdots \\ \vdots \\ 0 \end{pmatrix} \begin{pmatrix} 1 & \rho & \rho & \dots & \rho \\ \rho & 1 & \rho & \dots & \rho \\ \vdots & & \ddots & & \vdots \\ \rho & \rho & \dots & & 1 \end{pmatrix} \right)$$

where,

$$\begin{aligned} \sigma^2_{A(i,1)} \;&=\; \sigma^2_{A(i,2)} = \frac{(n-1)(2m-1)}{4} \\ U_j \;&=\; \frac{(A(i,j) + (n-1)/2)}{\sqrt{(n-1)(2m-1)/4}} \end{aligned}$$

$$\begin{aligned}\rho &= \frac{\mathbf{Cov}(A(i,1), A(i,2))}{(\sigma_{A(i,1)} \cdot \sigma_{A(i,2)})} \\ &= \frac{(m-2)}{(2m-1)}.\end{aligned}$$

Remark : For a fixed m and $\forall(\lambda_1, \ldots, \lambda_m) \in \mathbf{R}^m$, $(\lambda_1, \ldots, \lambda_m) \times \mathbf{Q}^{-\frac{1}{2}} \times [U_1, \ldots, U_m]$ approximately follows $\mathbf{N}(0, [\lambda_1^2 + \ldots + \lambda_m^2])$ as $n \to \infty$ where () and [] mean a row vector and a column vector respectively, and $\mathbf{Q}$ is the *variance-covariance matrix* of $U_1, \ldots, U_m$. Hence, by Cramer-Wold device [9, page 397], U_j's are approximately jointly normal.

Theorem 3.4 : The probability (i.e., $\{\mathbf{P}(U_1 < h, \ldots, U_m < h)\}^n$) that all of the m stored patterns are stable is

$$\left[\int_{-\infty}^{\infty} \Phi^m \left(\frac{(h - \sqrt{\rho} \cdot z_0)}{\sqrt{1-\rho}}\right) \cdot 1/\sqrt{2\pi} \cdot e^{-z_0^2/2} dz_0\right]^n \tag{3.15}$$

where $h = \frac{((n-1)/2)}{\sqrt{\sigma_{A(i,1)}}}$, and ρ, $U_j's$ are defined as before. Φ denotes the standard normal **CDF**.

Proof: Let $(Z_0, Z_1, \ldots, Z_m)$ be *i.i.d.* $\mathbf{N}(0,1)$ variables. In the following transformation (Gupta [34]) :

$$\begin{aligned}Y_1 &= \sqrt{\rho} Z_0 + \sqrt{1-\rho} Z_1, \\ Y_2 &= \sqrt{\rho} Z_0 + \sqrt{1-\rho} Z_2, \\ &\vdots \\ Y_m &= \sqrt{\rho} Z_0 + \sqrt{1-\rho} Z_m,\end{aligned}$$

$\mathbf{E}(Y_i) = 0, \mathbf{Var}(Y_i) = \rho + (1-\rho) = 1$, and $\mathbf{Cov}(Y_1, Y_2) = \rho$. Since the mean and the covariance of $(Y_1, \ldots, Y_m)$ are same as the mean and the covariance of $(U_1, \ldots, U_m)$,

and both have multivariate normal distribution, the distribution of $(U_1, \ldots, U_m)$ and $(Y_1, \ldots, Y_m)$ are identical. Therefore,

$$\begin{aligned}
&\mathbf{P}(U_1 < h, \ldots, U_m < h) = \mathbf{P}(Y_1 < h, \ldots, Y_m < h) \\
&= \mathbf{P}(\sqrt{\rho}Z_0 + \sqrt{1-\rho}Z_1 < h, \ldots, \sqrt{\rho}Z_0 + \sqrt{1-\rho}Z_m < h) \\
&= \mathbf{P}(Z_1 < \frac{(h - \sqrt{\rho}Z_0)}{\sqrt{1-\rho}}, \ldots, Z_m < \frac{(h - \sqrt{\rho}Z_0)}{\sqrt{1-\rho}}) \\
&= \int_{-\infty}^{\infty} \Phi^m \left(\frac{(h - \sqrt{\rho} \cdot z_0)}{\sqrt{1-\rho}} \right) \cdot \frac{1}{\sqrt{2\pi}} \cdot e^{-z_0^2/2} dz_0. \quad \square
\end{aligned}$$

3.5 Results

In this section, we present the numerical results based on our theoretical work and compare them with the results of other theoretical studies (Amari, Hopfield, McEliece, and Weisbuch). The simulation results of Weisbuch [96] are used as criteria for the comparison.

Weisbuch's simulation (refer to Appendix B) results are based on testing mn inequalities (invariance conditions, Equation (3.5) or Equation (3.6)) per network and by adjusting m so that the probability for all inequalities to be verified is 0.5. We and Weisbuch study the information capacity under the condition that the probability that all of the m stored patterns are stable is 0.5 (i.e., $\mathbf{I}_{0.5}$ in Definition 3.4). On the other hand, Amari, Hopfield, and McEliece used the condition that *most* of m stored patterns are stable, but the definition of *most* is left ambiguous in their study. The comparison among Weisbuch's simulation study and the various theoretical studies is summarized in Table 3.1.

The theoretical results of Hopfield [43], Amari [6] and McEliece [69] overestimate the information capacity in comparison to the simulation study of Weisbuch [96]. Our

Table 3.1: The required number of neurons n for storing m patterns

Methods	m = number of patterns					
	8	10	14	20	1,000	50,000
Hopfield	53	67	93	133	6,667	333,333
Amari	20	28	43	69	7,120	534,939
McEliece	21	30	47	76	7,783	576,046
Weisbuch (with assumptions)	149	210	343	562	62,528	4,737,757
LKS (ours)	146	206	337	552	61,842	4,698,517
Weisbuch's Simulation	103	180	290	500	N/A	N/A

results and Weisbuch's theoretical results (using Assumption 3.2 and Assumption 3.3) are close to the results of the simulation. The overestimation occurs because the studies [7, 43, 69] are based on the condition that a stored pattern is stable, instead of stability of m patterns. In any case except Hopfield's theoretical dynamics, the ratio of m to n converges to 0, which means the information capacity is equal to 0 in the exact sense. The asymptotic behavior ofthe information capacity is described in Figure 3.2.

The comparison between our and Weisbuch's results is provided in Table 3.2. Our analysis reduces almost 12% of the discrepancy between the simulation results and Weisbuch's theoretical results.

Table 3.2: Comparison with Weisbuch's results

Items	m = number of patterns			
	8	10	14	20
(A) Simulation	103	180	290	500
(B) Weisbush	149	210	343	562
(C) LKS (ours)	146	206	337	552
(1 - (C - A)/(B - A)) x 100 = gap reduction	6.65%	13.33%	11.32%	16.13%

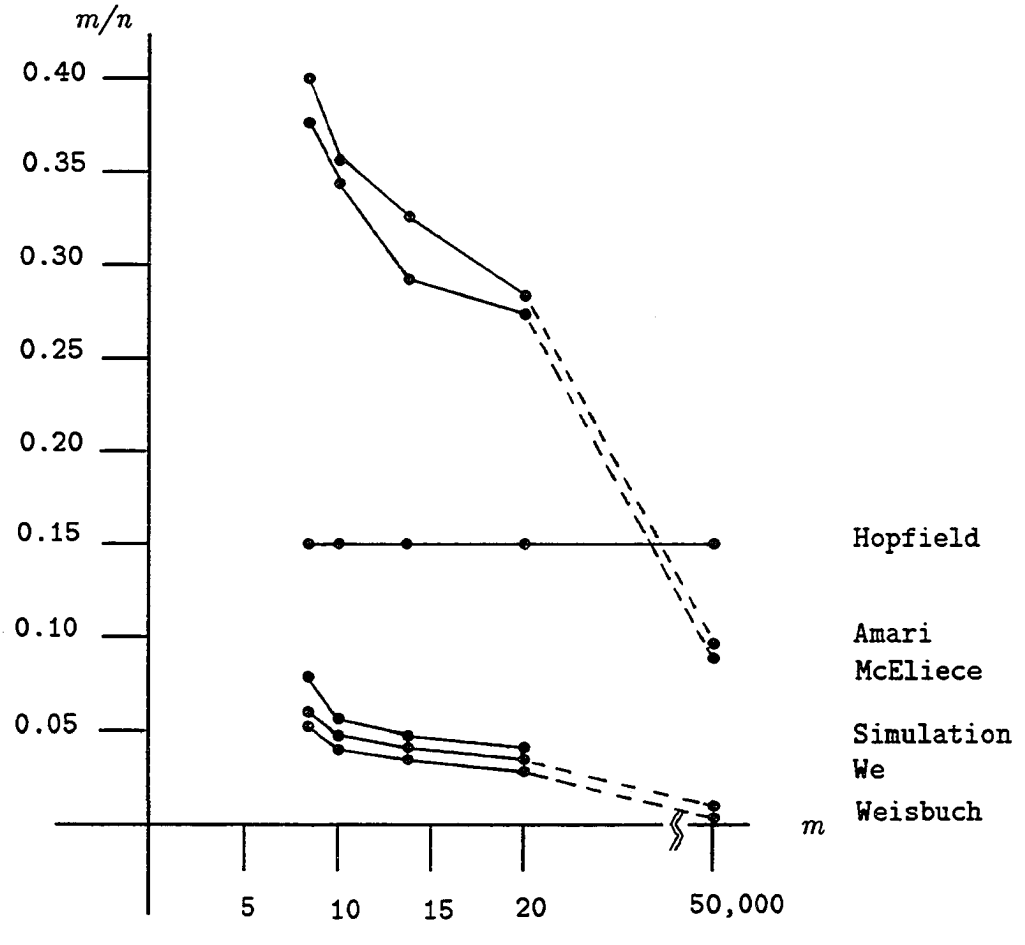

Figure 3.2: Asymptotic behavior of the information capacity

3.6 Concluding Remarks

We establish the connection between the dynamics of Hopfield Neural Network (HNN) and the theory of multivariate normal distribution and use the connection to derive the results about information capacity of HNN. The current information capacity heuristics due to Amari, Hopfield, and McEliece and Weisbuch are carefully reviewed, and the sources of inaccuracies are examined. We have been successful in resolving most of the inaccuracies. This has mainly been possible because of some nontrivial results from statistics due to Brown and Gupta [11, 34].

Based on our analysis and using the statistical software package **IMSL** [49], we derive numerical results for the information capacity of the Hopfield network. Our results are closer to the simulation [96] than the results from other theoretical studies [6, 7, 43, 69, 96]. Because of high computational requirements it is difficult to do simulations of networks with large number of neurons. Simulation results in [96], as shown in Table 3.1, only go up to neural networks of 500 neurons. Our analysis can be applied to the study of networks of large size.

Our research along with other theoretical studies of capacity heuristics can be put in the perspective by considering the following four categories. Note that the quantity $\mathbf{P}(\mathbf{A}(i,s) < 0)$, as defined in Equation (3.9), is the probability that a given neuron satisfies the invariance condition. Each category defines a different way of calculating the probability that all m patterns are stable.

Category 1: $\{\mathbf{P}(\mathbf{A}(i,s) < 0)$, for a fixed i and a fixed $s\}^{nm}$; no consideration of any dependency among the terms $-S_i^s + N_i^s$, $\forall\, i, s \;\; 1 \leq i \leq n,\; 1 \leq s \leq m$.

Category 2: $\{\mathbf{P}(\mathbf{A}(i,s) < 0)$, $\forall i = 1, \ldots, n$, and for a fixed $s\}^{m}$; consideration

of the dependency among the terms $-S_i^s + N_i^s$ for n neurons in a pattern.

Category 3: $\{\mathbf{P}(\mathbf{A}(i,s) < 0), \quad \forall s = 1,\ldots,m, \text{ and for a fixed } i\}^n$; consideration of the dependency among the terms $-S_i^s + N_i^s$ for i^{th} neurons in m stored patterns.

Category 4: $\{\mathbf{P}(\mathbf{A}(i,s) < 0, \quad \forall\, i,s, \quad i = 1,\ldots,n, \text{ and } s = 1,\ldots,m\}$; consideration of every dependency.

Most researchers have tried to derive a mathematical expression for the information capacity based on Category 1 using Assumption 3.2 and Assumption 3.3. By eliminating Assumption 3.2 by using Brown's Martingale Central Limit Theorem, we refine the current capacity heuristics. In addition to that, by eliminating Assumption 3.3 by using Gupta's theorem, we are able to extend the mathematical analysis to estimate the information capacity based on Category 2 and Category 3. The estimation of the information capacity based on Category 4 remains an open problem.

4. INTERACTIVE LOGICAL DATABASE WITH MACRO-NEURONS BASED INFERENCE ENGIN

4.1 Introduction

Since Hopfield [43] rekindled the interest in networks of automata by introducing a new kind of associative memory based on a simple neural network model, several trials for pursuit of the ability of recalling memorized patterns of human memory have been done.

Both Cernuschi-Frias [14] and Goles [30] generalized the Hopfield memory model by considering partition of the network into blocks and simultaneous update of units (neurons) in each block. Especially, Cernuschi-Frias viewed a block as a macro-neuron and hinted that the notion of macro-neurons can be used for higher level computation, for example, cognitive process. Some relaxation on the symmetry hypothesis of the weight matrix was also discussed by both of them. At the same time, Kosko [60] introduced a bidirectional heteroassociative Content Addressable Memory (CAM) called Bidirectional Associative Memory (BAM) and exhibited the possibility of storing temporal patterns in a BAM. BAM can be thought as Block Hopfield Memory (BHM) with two blocks.

In the field of cognitive science, McClelland [76] developed a simulation model of neural network as a CAM. In fact, the result of the simulation showed that the model

was considerably more powerful than a conventional CAM. It performed the following tasks: default assignment, graceful degradation, and spontaneous generalization.

On the other hand, recently there has been increasing interest in the application of neural networks as useful tools for Artificial Intelligence (AI) tasks. Several neural network architectures that provide a distributed representation for frame-like concept structure have been presented in [15, 21, 50, 87, 88, 94]. These neural network architectures were considered to develop a powerful short-term memory that can construct and manipulate concepts or symbolic schemata rapidly. In other words, in this area neural networks are used mainly for brain modeling.

In expert system area , [28, 38, 42, 47, 77, 88] neural networks are also used for implementing knowledge acquisition task which is a main bottleneck and for replacing logical inference and unification process to use adaptability, speed, and robustness of neural networks.

Gallant [28] set up the basic structure for embedding a neural network architecture in expert systems. But no notion of Short-Term Memory (STM) was used in his work. Hollbach's immediate inference and mediated inference [42] are corresponding to default assignment and spontaneous generalization respectively which are generic characteristics of Hopfield associative memory as mentioned by McClelland [67]. And Samad's chained inference rule [77] can be compared to storing temporal patterns into a BAM.

The combination of these research activities allows us to construct an useful experimental database system what we call Interactive Logical Database with Macro-Neurons Based Inference Engine. We follow Hillman's basic strategies [38] in integrating an expert system and a neural network. Our attempt is to exploit advantages

of various approaches and to combine them into a useful unified approach. The main subsystems of ILDBMN and the basic idea for their implementation are summarized in Table 4.1.

Table 4.1: The basic structure of ILDBMN and origins of implementation idea

Subsystems	Language	Idea
USERIO, USERINTERFACE	Prolog	* Problem domain independent inference engine by separating these parts
NEUROINFERENCE	Prolog	* **BHM**: low level inference * **BAM**: concept level inference
BINDER	Prolog	* Simple role binding * Convenient communication between external world and inference engine
WEIGHTS	Prolog	* Expert knowledge base as a correlation matrix
LEARNING	C	* Modified Hebbian learning * Relaxation of the symmetry condition on weight matrix

4.2 Theoretical Background

The main subsystems of ILDBMN are NEUROINFERENCE, LEARNING, and WEIGHTS which are in undetachable relation. For clarity we briefly review the theoretical background and the idea for combining scattered research activities.

The updating rule for each neuron in the original Hopfield neural network model [43] is

$$x_i^{'} = sgn\left\{\sum_{j=1}^{N} w_{i,j} \cdot x_j\right\}, \tag{4.1}$$

where N is the number of neurons, $W = \{w_{i,j}\}$ is a weight matrix generated by Hebbian learning, x_i has the activation value $+1$ or -1, and sgn (y) $= \left\{ \begin{array}{ll} +1 & \text{if } y \geq 0 \\ -1 & \text{otherwise} \end{array} \right\}$.

Cernuschi-Frias [14] and Goles [30] generalized the Hopfield model by considering updating simultaneously a block of neurons. These blocks are disjoint in the sense that each neuron belongs to only one block. The generalization is obtained as follows: i^{th} block is updated according to

$$\mathbf{X}_i^{'} = sgn\left\{\sum_{j=1}^{M} \mathbf{W}_{i,j} \cdot \mathbf{X}_j\right\}, \tag{4.2}$$

where M is the number of blocks, $\mathbf{X}_i$ is a column vector corresponding to a block of q_i neurons, each taking value $+1$ or -1, and $\mathbf{W}_{i,j}$ is a $q_i \times q_j$ matrix of weights for connections with another block of q_j neurons.

Specifically each neuron in a block is updated according to

$$x_i^{'}(k) = sgn\left\{\sum_{j=1}^{M}\sum_{h=1}^{q_j} \mathbf{W}_{i,j}(k,h) \cdot x_j(h)\right\}, \tag{4.3}$$

where $x_i^{'}(k)$ means new activation value of k^{th} neuron in the i^{th} block and $\mathbf{W}_{i,j}(k,h)$ means $(k,h)^{th}$ element in $\mathbf{W}_{i,j}$.

Theorem 4.1 : Let's define the energy function **E** for a BHM to be

$$\mathbf{E} = -\sum_{i=1}^{M}\sum_{j=1}^{M} \mathbf{X}_i^{'} \cdot \mathbf{W}_{i,j} \cdot \mathbf{X}_j^{'}. \tag{4.4}$$

If blocks are chosen at random one after another and each block is updated according to Equation (4.2) and Equation (4.3), then the BHM will be stabilized at the local minima of the energy function **E** within polynomial number of cycles with respect to the number of blocks. *Refer to Cernuschi-Frias [14] and Goles [30] for stabilization, and Abu-Mostafa [1] and Goles [30] for polynomial time complexity.*

Let's consider the example of a BHM in Figure 4.1 We can think each block

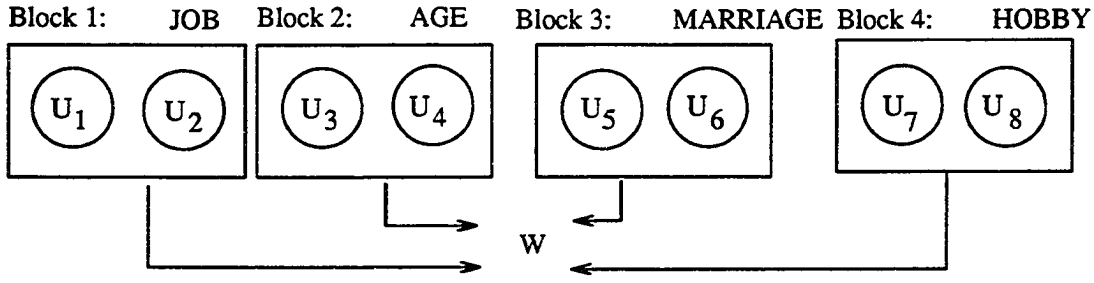

Figure 4.1: An example of 4-block BHM

of a macro-neuron which has multivalued states. For example, the second block may have a role of attribute AGE in a database, and four possible activation states can be values such as 20, 30, 40, and 50. Hence, slot-filler structure in symbolic schemata can be easily embedded in a BHM by mapping slot and filler to block (attribute) and activation state respectively.

Independently, Kosko [60] introduced a BAM behaving as a heteroassociative CAM, which can store temporal patterns. In our research, we use BAM for storing temporal patterns for chained inference or concept level inference.

In a BAM, a temporal pattern $(+1,-1,-1,-1)$ $(X_1) \rightarrow (-1,+1,+1,-1)$ (X_2) $\rightarrow (-1,-1,-1,+1)$ (X_3) can be stored as in Figure 4.2. In the BAM, if we activate block 1 with X_1, we get X_2 and if activate with X_2, we get X_3.

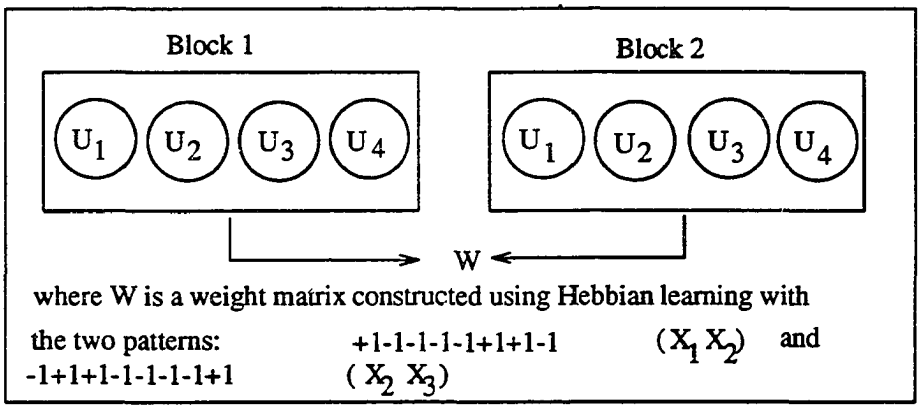

Figure 4.2: An example of storing a temporal pattern in a BAM

Consider a database of four attributes such as JOB, AGE, MARRIAGE, and HOBBY. The concept level relational information among attributes, for example, JOB $\rightarrow$ AGE, MARRIAGE, and AGE, MARRIAGE $\rightarrow$ HOBBY, where A $\rightarrow$ B means that the attribute B is a generalized characteristics of the attributes A, can easily be saved in a BAM as follows:

- Construct a BAM with 4 units in each block.
- The first unit in each block is for the attribute JOB, the second unit for AGE, the third unit for MARRIAGE, and the fourth unit for HOBBY.
- Storing conceptual relationships such as JOB $\rightarrow$ AGE, MARRIAGE and AGE, MARRIAGE $\rightarrow$ HOBBY amounts to storing patterns in the BAM as in Figure 4.2 so that $(+1,-1,-1,-1)$ in block 1 produces $(-1,+1,+1,-1)$ as the

output in block 2 and similarly $(-1,+1,+1,-1)$ produces $(-1,-1,-1,+1)$ as the output.

Hence, if we have the information *JOB: student*, then first using a four blocks (JOB, AGE, MARRIAGE, HOBBY) BHM, we can get the information about each attribute from its stable state. Next, using a BAM we can get a concept level relational information generated from the given input data *JOB: student.* For example, in the four blocks BHM, the remaining attributes will be filled in with some values, let's say, *AGE: 20, MARRIAGE: single,* and *HOBBY: football.* Then by using the BAM, the system will automatically search the higher level concept generated from the given input data *Job: student.* More specifically, we get the information that student's AGE is generally 20 and their MARRIAGE state is single (recall that JOB $\rightarrow$ AGE, MARRIAGE) first and next, we get the information that their HOBBY is football (recall that AGE, MARRIAGE $\rightarrow$ HOBBY).

Until now, the basic theoretical background for NEUROINFERENCE is discussed. The important next step in understanding ILDBMN is to select an appropriate LEARNING algorithm producing WEIGHTS matrix for NEUROINFERENCE.

As a by-product of the proof for Theorem 4.1, Cernuschi-Frias [14] and Goles [30] proved also that diagonal matrices of the weight matrix of a BHM don't need to be symmetric. For example, a BHM with four units and two blocks, can have the weight matrix as shown in Figure 4.3, where $\mathbf{W}_{1,1}$ and $\mathbf{W}_{2,2}$ need not to be 0-diagonal symmetric matrices as in the original Hopfield associative memory model.

If $\mathbf{W}_{i,i}$ where $1 \leq i \leq M$ and M is the number of blocks, is *a nonnegative definite matrix*, stability of the BHM is guaranteed. Using this result, we modify the Hebbian learning as follows:

Step 1: Apply the Hebbian learning for all input patterns.

Step 2: Change the diagonal weight matrix ($\mathbf{W}_{i,i}$) to a diagonal dominant matrix, i.e., $\mathbf{W}_{i,i}(k,k) \geq \sum_{l \in q_i, l \neq k} \left|\mathbf{W}_{i,i}(k,l)\right|$ which is a sufficient condition for *non negative definite matrix.*

An example of a weight matrix obtained by Hebbian learning from input patterns is provideed in (a) of Figure 4.4. Then by Step 2, the matrix will be changed as in (b) of Figure 4.4.

	Block 1		Block 2	
	U_1	U_2	U_3	U_4
U_1	W_{11}		W_{12}	
U_2				
U_3	W_{21}		W_{22}	
U_4				

Figure 4.3: An example of a weight matrix for a four units BHM of two blocks

	U_1	U_2	U_3	U_4
U_1	(0)	3	3	5
U_2	3	(0)	-4	-2
U_3	3	-4	(0)	-2
U_4	5	-2	-2	(0)

(a) Step 1

⇒

	U_1	U_2	U_3	U_4
U_1	(3)	3	3	5
U_2	3	(3)	-4	-2
U_3	3	-4	(2)	-2
U_4	5	-2	-2	(2)

(b) Step 2

Figure 4.4: A weight matrix of a BHM based on the modified Hebbian learning

This learning algorithm has some advantages over the original Hebbian learning under special encoding scheme of external information. Let's consider the specific example in Figure 4.5 of an encoding scheme for an external information. Here, as soon as unit 1 is activated, i.e., Lee, unit 1 strongly tends to keep *on* state (+1) inside of block 1 because of diagonal dominant condition in weight matrix. This characteristic of the new learning algorithm has a momentum effect for convergence in updating process.

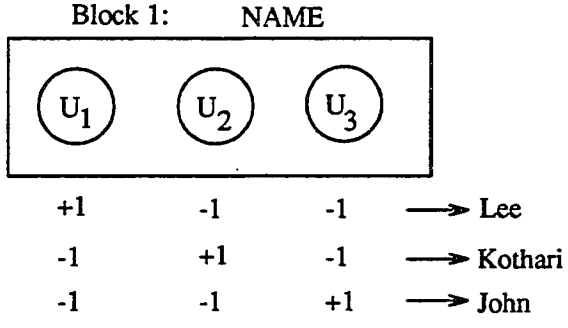

Figure 4.5: A specific encoding scheme for the Block 1: NAME

4.3 Overview of ILDBMN

Usually, expert systems supply a flexible user interface and work well for domain specific problems. On the other hand, neural networks are very powerful in providing general classification of a set of inputs [38].

ILDBMN is an expert database system constructed by integrating expert system techniques and neural networks to exploit advantages of both techniques. ILDBMN consists of five subsystems: USERIO, USERINTERFACE, NEUROINFERENCE, BINDER, WEIGHTS, and LEARNING. The relationship among subsystems is summarized in Figure 4.6 and the main function of each subsystem is described below.

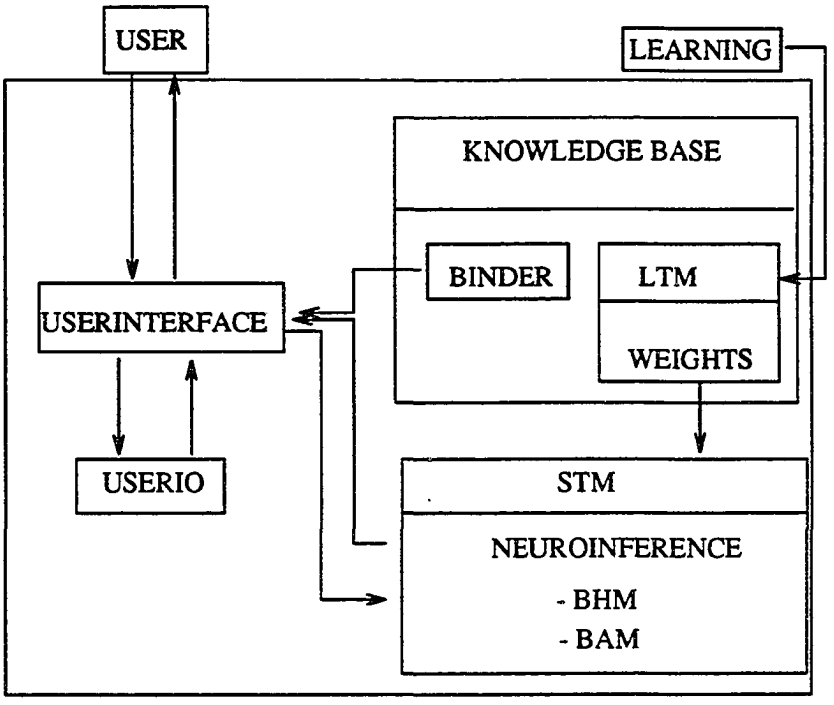

Figure 4.6: The relationship among subsystems in ILDBMN

USERIO, USERINTERFACE: USERIO provides userfriendly interactive communication environment through several menus and leading messages. USERINTERFACE delivers external input data from USERIO to NEUROINFERENCE for decision making and returns the results from NEUROINFERENCE to user through USERIO. For example, from the input data generated from USERIO such as *JOB: student, AGE: no data, EDUCATION: no data* USERINTERFACE provides NEUROINFERENCE with a list [1, 0, 0] which means that first unit in the first block is in the *on* state as an external input and we don't know anything about the remaining two blocks.

NEUROINFERENCE: Using an input list from USERINTERFACE, i.e., [1, 0, 0],

NEUROINFERENCE runs two different neural networks to search for required information using WEIGHTS which is a kind of Long Term Memory (LTM), storing known information. One neural network is a BHM which is used to fill in the unfilled block of input from USERIO. The other is a BAM which is used to extract higher level concept. After two neural networks become stable, NEUROINFERENCE returns two list as results to USERINTERFACE. For example, With the input [1, 0, 0] under the encoding scheme as in Figure 4.7, NEUROINFERENCE may return the results [1, 1, 2] from the BHM and [2, 3] from the BAM based on the WEIGHTS matrix. The list [1, 1, 2] means *JOB: student, AGE: 20,* and *MARRIAGE: single* and the list [2, 3] means JOB $\rightarrow$ AGE, MARRIAGE as defined in the background section.

BINDER: BINDER is used for convenient communication between external world and NEUROINFERENCE. The output lists from NEUROINFERENCE are interpreted by USERINTERFACE for the user by using BINDER such as in Figure 4.8.

LEARNING, WEIGHTS: WEIGHTS matrix stores an expert's knowledge or external known information as a correlation matrix generated by LEARNING as described in the background section. The connection $w_{i,j} = 5$ is represented by the fact $weight(i, j, 5)$ in Prolog.

A full example of sample run of ILDBMN is shown in Appendix C.

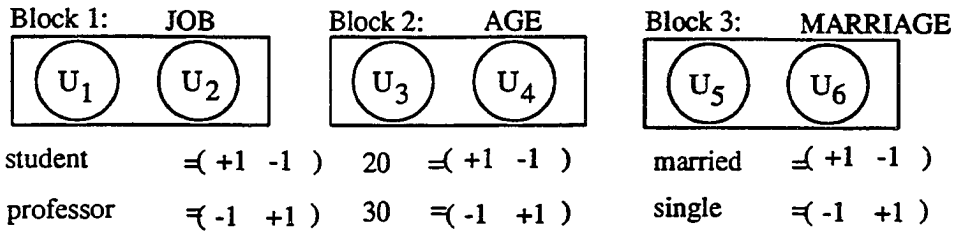

Figure 4.7: An example of an encoding scheme

job(1, student)	attributename(1, job)
job(2, professor)	attributename(2, age)
age(1, 20)	attributename(3, marriage)
age(2, 30)	
marriage(1, married)	
marriage(2, single)	

Figure 4.8: An example of BINDER

4.4 Concluding Remarks

ILDBMN suggests an elegant way for implementing a flexible, robust database based on AI technique of separating the inference engine from database for problem-domain independent implementation, and a neural network architecture based on the notion STM and LTM.

By using a neural network, ILDBMN can provide speedy, robust decision making and perform adaptable knowledge acquisition task which is a main bottleneck in building expert systems. On the other hand, ILDBMN can provide userfriendly

interactive communication environment by keeping general expert system structure. ILDBMN can naturally be used for relational database, logic database, and expert systems for decision making.

5. CONCLUSIONS

In this research work, we provide a global picture of the current state of the art of neural networks in a consistent way. By doing so, a guide for beginner in the field of neural networks is provided and the starting point of our main researches is clearly set up in the global picture.

As a theoretical study, a probabilistic definition of information capacity of the Hopfield Associative Memory is formalized by introducing strong statistical tools for analyzing the capacity of Hopfield neural network. With the tools, we mathematically justify several assumptions which have important roles in current information capacity heuristics. The statistical tools will also suggest a way for estimating the information capacity of neural networks of large size in which the estimation of the capacity can't be done by computer simulation because of time complexity.

From the viewpoint of application, we develop an useful experimental logical expert database system called *Interactive Logical Database System with Macro-neurons Based Inference Engine* by combining well-known theoretical results of Hopfield neural network with research activities from the field of AI and cognitive science.

5.1 Future Direction

Through this research work, we have tried to do a systematic and consistent study of neural networks by following the line *Survey* $\longrightarrow$ *Theoretical Study* $\longrightarrow$ *Study for Applications* under a subject, i.e., Hopfield neural network. To improve the consistency in following the line, future work would study the following areas.

1. Theoretical study of information capacity of Hopfield neural network for structured patterns.

2. Performance analysis to study information capacity of BHM and efficiency of the inference mechanism in ILDBMN.

3. Implementation of run-time LEARNING in ILDBMN.

4. Development of a better encoding scheme for reducing the number of units of BHM in ILDBMN.

6. BIBLIOGRAPHY

[1] Y. S. Abu-Mostafa. "Neural Networks for Computing." *AIP Conference Proceedings* 151. Snowbird, UT: AIP, 1987.

[2] Y. S. Abu-Mostafa. "Connectivity versus Entropy." *Neural Information Processing Systems*, Ed. D.Z. Anderson. New York, NY: AIP, 1988. 1–8.

[3] Y. S. Abu-Mostafa and J. S. Jacques. "Information Capacity of the Hopfield Model." *IEEE Trans. on IT* 31, No. 4 (1985): 461–464.

[4] D. H. Ackley and G. E. Hinton. "A Learning Algorithm for Boltzmann Machines." *Cognitive Science* 9 (1985): 147–169.

[5] I. Aleksander. *Neural Computing Architectures: The Design of Brain-Like Machines*, Cambridge, MA: The MIT Press, 1989.

[6] S. Amari. "On Mathematical Methods in the Theory of Neural Networks." *IEEE International Conference on Neural Networks Vol. III.* San Diego, CA: SOS Printing, 1987.

[7] S. Amari, K. Yoshida, and K. Kanatani. "A Mathematical Foundation for Statistical Neurodynamics." *SIAM Journal of Applied Math.* 33, No. 1 (1977): 95–126.

[8] J. A. Anderson and E. Rosenfeld. *Neurocomputing: Foundations of Research*, Cambridge, MA: The MIT Press, 1988.

[9] P. Billingsley. *Probability and Measure*, New York, NY: John Wiley & Sons, 1986.

[10] L. J. Brain. *Introduction to Probability and Mathematical Statistics*, Boston, MA: Duxbury Press, 1987.

[11] B. B. Brown. "Martingale Central Limit Theorems." *The Annals of Mathematical Statistics* 42, No. 1 (1971): 59–66.

[12] J. Bruck and J. W. Goodman. "A Generalized Convergence Theorem for Neural Networks and its Applications in Combinatorial Optimization." *IEEE International Conference on Neural Networks Vol. III.* San Diego, CA: SOS Printing, 1987.

[13] G. A. Carpenter. "Neural Network Models for Pattern Recognition and Associative Memory." *Neural Networks* 2 (1989): 243–257.

[14] B. Cernuschi-Frias. "Partial Simultaneous Updating in Hopfield Memories." *IEEE Trans. On SMC* 19, No. 4 (1989): 887–888.

[15] H. W. Chun and A. Mimo. "A Massively Parallel Model of Schema Selection." *IEEE International Conference on Neural Networks Vol. II.* San Diego, CA: SOS Printing, 1987.

[16] M. Cottel. "Stability and Attractivity in Associative Memory Networks." *Biological Cybernetics* 58 (1988): 129–139.

[17] E. D. Dahl. "Neural Networks Algorithm for an NP-Complete Problem: Map & Graph Coloring." *IEEE International Conference on Neural Networks Vol. III.* San Diego, CA: SOS Printing, 1987.

[18] *DARPA Neural Network Study*, Fairfax, VA: AFCEA International Press, 1988.

[19] E. David and T. K. Miller. "A Traveling Salesman Objective Function that Works." *IEEE International Conference on Neural Networks Vol. II.* San Diego, CA: IEEE TAB Neural Network Conference, 1988.

[20] G. W. Davis. "Sensitivity Analysis in Neural Net Solutions." *IEEE Trans. on SMC* 19, No. 5 (1989): 1078–1082.

[21] M. Derthick. "A Connectionist Architecture for Representing and Reasoning about Structured Knowledge." *Program of the Ninth Annual Conference of the Cognitive Science Society.* Seattle, WA: Cognitive Science Society, 1987.

[22] C. P. Dolan and M. G. Dyer. "Symbolic Schemata, Role Binding, and the Evaluation of Structure in Connectionist Memories." *IEEE International Conference on Neural Networks Vol. II.* San Diego, CA: SOS Printing, 1987.

[23] S. E. Fahlman, G. E. Hinton, and T. J. Sejnowski. "Massively Parallel Architecture for AI: NETL, Thistle, and Boltzmann Machines." *Proceedings of the National Conference on Artificial Intelligence.* Washington, D.C.: AAAI-83, 1983.

[24] J. A. Feldeman, M. A. Fanty, N. H. Goddard, and K. J. Lynne. "Computing with Structured Connectionist Networks." *Communications of the ACM* 31 (1988): 170–187.

[25] M. Fleisher. "The Hopfield Model with Multi-Level Neurons." *Neural Information Processing Systems*, Ed. D.Z. Anderson. New York, NY: AIP, 1988. 278–289.

[26] F. S. Fogelman, E. Goles, and G. Weisbuch. "Transient Length in Sequential Iteration of Threshold Functions." *Discrete Applied Mathematics* 6 (1983): 95–98.

[27] F. S. Fogelman, Y. Robert, and M. Tchuente. *Automata Networks in Computer Science: Theory and Application*, Princeton, NJ: Princeton University Press, 1987.

[28] S. I. Gallant. "Connectionist Expert Systems." *Communications of the ACM* 31, No. 2 (1988): 52–69.

[29] E. Goles. "Fixed Point Behavior of Threshold Functions on a Finite Set." *SIAM J. Alg. Disc. Math.* 3, No. 4 (1982): 529–531.

[30] E. Goles. "Decreasing Energy Functions as a Tool for Studying Threshold Networks." *Discrete Applied Mathematics* 12 (1985): 261–277.

[31] E. Goles. "Dynamics of Positive Automata Networks." *Theoretical Computer Science* 41 (1985): 19–32.

[32] E. Goles. "Local Graph Transformations Driven by Lapunov Functionals." *Complex Systems* 3 (1989): 173–184.

[33] E. Goles and S. Martinez. "Properties of Positive Functions and the Dynamics of Associated Automata Networks." *Discrete Applied Mathematics* 18 (1987): 39–46.

[34] S. S. Gupta. "Probability Integrals of Multivariate Normal and Multivariate *t*." *The Annals of Mathematical Statistics* 34, No. 3 (1963): 792–828.

[35] G. Hadley. *Linear Algebra*, Reading, MA: Addison-Wesley Publishing Company, 1972.

[36] K. Hanies and R. Hecht-Nielsen. "A BAM with Increased Information Storage Capacity." *IEEE International Conference on Neural Networks Vol. I.* San Diego, CA: IEEE TAB neural Network Conference, 1988.

[37] S. U. Hegde, J. L. Sweet, and W. B. Levy. "Determination of Parameters in a Hopfield/Thank Computational Network." *IEEE International Conference on Neural Networks Vol. II.* San Diego, CA: IEEE TAB Neural Network Conference, 1988.

[38] D. V. Hillman. "Integrating Neural Nets and Expert Systems." *AI Expert* (June 1990): 54–59.

[39] G. E. Hinton and J. A. Anderson. *Parallel Models of Associative Memory*, Hillsdale, NY: Lawrence Erlbaum Associates, 1981.

[40] G. E. Hinton and T. J. Sejnowski. "Analyzing Cooperative Computation." *Proceedings of the Fifth Annual Conference of the Cognitive Science Society.* Rochester, NY: Cognitive Science Society 1983.

[41] R. V. Hogg and A. T. Graig. *Introduction to Mathematical Statistics*, New York, NY: Macmillan Publishing Co., 1978.

[42] S. C. Hollbach. "Direct Inference in a Connectionist Knowledge Structure." *Proceedings of the Tenth Annual Meeting of the Cognitive Science Society*, 1988.

[43] J. J. Hopfield. "Neural Networks and Physical Systems with Emergent Collective Computational Abilities." *Proceedings of the National Academy of Science U.S.A.* 79 (1982): 2554–2558.

[44] J. J. Hopfield. "Neurons with Graded Response Have Collective Computational Properties like Those of Two-State Neurons." *Proceedings of the National Academy of Science U.S.A.* 81 (1984): 3088–3092.

[45] J. J. Hopfield. "Artificial Neural Network." *IEEE Circuit and Device Magazine* 4, No. 5 (1988): 3–10.

[46] J. J. Hopfield and D. W. Tank. "Neural Computation of Decision in Optimization." *Biological Cybernetics* 52 (1985): 141–152.

[47] M. N. Huhns. *Distributed Artificial Intelligence*, Los Altos, CA: Kaufmann Publisher, 1987.

[48] L. P. Hyvärinen. *Information Theory for Systems Engineers*, Berlin, Heidelberg: Springer-Verlag, 1970.

[49] *IMSL User's Manual Math/Liabrary Fortran Subroutines for Mathematical Applications, Version 1.0*, Huston, TA: Problem-Solving Software System, 1987.

[50] A. Jagota and O. Jakubowich. "Knowledge Representation in a Multi-Layered Hopfield Network." *IEEE ICNN International Joint Conference on Neural Networks Vol. I.* San Diego, CA: IEEE TAB Neural Network Conference, 1989.

[51] G. Josin. "Integrating Neural Networks with Robotics." *AI Expert* (August 1988): 50–58.

[52] G. Josin. "Neural-Space Generalization of a Topological Transformation." *Biological Cybernetics* 59 (1988): 283–290.

[53] S. Judd. "Learning in Networks is Hard." *IEEE International Conference on Neural Networks Vol. II.* San Diego, CA: SOS Printing, 1987.

[54] S. Judd. "On the Complexity of Loading Shallow Neural Networks." *Journal of Complexity* 4 (1988): 177–192.

[55] J. D. Keeler. "Basins of Attraction of Neural Network Model." *AIP Conference Proceedings* 151, Snowbird, UT: AIP, 1986.

[56] S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi. "Optimization by Simulated Annealing." *Science* 220, No. 4598 (1983): 671–680.

[57] T. Kohonen. *Associative Memory: A System-Theoretical Approach*, Berlin, Heidelberg: Springer-Verlag, 1977.

[58] B. Kosko. "Adaptive Bidirectional Associative Memories." *Applied Optics* 26, No. 23 (1987): 4947–4960.

[59] B. Kosko. "Constructing an Associative Memory." *Byte* (September 1987): 137–144.

[60] B. Kosko. "Bidirectional Associative Memories." *IEEE Trans. on SMC* 18, No. 1 (1988):49–60.

[61] A. Krowitch, L. Rendell, and B. Hohensee. "The State Space of Memory Clusters in the Hopfield Networks." *IEEE International Conference on Neural Networks Vol. III.* San Diego, CA: SOS Printing, 1987.

[62] S. Y. Kung and J. N. Hwang. "A Unified Architecture for Artificial Intelligence." *Journal of Parallel and Distributed Computing* 6 (1989): 358–387.

[63] K. Lee and S. C. Kothari. "Interactive Logical Database with Macro-Neurons Based Inference Engine." *IASTED International Conference on Expert Systems and Neural Networks.* Honolulu, HI, 1990.

[64] R. P. Lippmann. "An Introduction to Computing with Neural Nets." *IEEE ASSP Magazine* (April 1987): 4–22.

[65] H. G. Loos. "Reflexive Associative Memories." *Neural Information Systems*, Ed. D. Z. Anderson. New York, NY: AIP, 1988. 495–504.

[66] P. K. Mazaika. "A Mathematical Model of the Boltzmann Machine." *IEEE International Conference on Neural Networks Vol. III.* San Diego, CA: SOS Printing, 1987.

[67] J. L. McClelland and D. E. Rumelhart. *Explorations in Parallel Distributed Processing: A Handbook of Models, Programs, and Exercise*, Cambridge, MA: The MIT Press, 1988.

[68] J. L. McClelland, D. E. Rumelhart, and The PDP Research Group. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Vol. 2. Psychological and Biological Models*, Cambridge, MA: The MIT Press, 1986.

[69] R. J. McEliece, E. C. Posner, E. R. Rodemich, and S. S. Venkatesh. "The Capacity of the Hopfield Associative Memory." *IEEE Trans. on IT* 33, No. 4 (1987): 461–482.

[70] M. Minsky and S. Papert. *Perceptrons, Expanded Edition*, Cambridge, MA: The MIT Press, 1988.

[71] E. W. Page and G. A. Tagliarini. "Algorithm Development for Neural Networks." *SPIE High Speed Computing* 880 (1988): 11–19.

[72] D. Prados and S. Kak. "Non-Binary Neural Networks." *Lecture Note in Control and Information Science* 130 (1989): 97–103.

[73] J. Ramanujam and P. Sadayappan. "Optimization by Neural Networks." *IEEE International Conference on Neural Networks Vol. II.* San Diego, CA: IEEE TAB Neural Network Conference, 1988.

[74] F. Robert. *Discrete Iterations: A Metric Study*, Berlin, Heidelberg: Springer-Verlag, 1986.

[75] M. G. Robert and C.C. Fernando. *Engineering Intelligent Systems: Concepts, Theory, and Application*, Bedford, MA: Digital Press, 1980.

[76] D. E. Rumelhart, J. L. McClelland, and The PDP Research Group. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Vol. 1. Foundations*, Cambridge, MA: The MIT Press, 1986.

[77] T. Samad. "Towards Connectionist Rule-Based Systems." *IEEE International Conference Neural Networks Vol. II.* San Diego, CA: IEEE TAB Neural Network Conference, 1988.

[78] R. Schank and R. Abelson. *Scripts, Plans, Goals, and Understanding*, Hillsdale, NJ: Lawrence Erlbaum Associates, 1977.

[79] P. K. Simpson. *Artificial Neural Systems: Foundations, Paradigms, Applications, and Implementations*, New York, NY: Pergamon Press, 1990.

[80] F. F. Soulie. "Parallel and Sequential Computation on Boolean Networks." *Theoretical Computer Science* 40 (1985): 275–300.

[81] F. F. Soulie. "Lapunov Functions and Their Use in Automata Networks." in *NATO ASI Series Vol. F20, Disordered Systems and Biological Organization*, pp. 85–100, Berlin, Heidelberg: Springer-Verlag, 1986.

[82] M. C. Stinson and S. C. Kak. "Bicameral Neural Computing." *Lecture Note in Control and Information Science* 130 (1989): 85–96.

[83] A. Stolcke. "Unification as Constraint Satisfaction in Structured Connectionist Networks." *Neural Computation* 1 (1989): 559–567.

[84] G. A. Tagliarini and E. W. Page. "Solving Constraint Satisfaction Problems with Neural Networks." *IEEE International Conference on Neural Networks Vol. III.* San Diego, CA: SOS Printing, 1987.

[85] Bev, Thompson and Bill, Thompson. "Neurons, Analog Circuit, and the Traveling Salesperson." *AI Expert* (July 1987): 28–32.

[86] D. S. Touretzky. "Symbols among the Neurons: Details of a Connectionist Inference Architecture." *Proceedings of Ninth International Joint Conference on Artificial Intelligence*, Los Angeles, CA, 1985.

[87] D. S. Touretzky. "Representing Conceptual Structures in a Neural Network." *IEEE International Conference on Neural Networks Vol. II.* San Diego, CA: SOS Printing, 1987.

[88] D. S. Touretzky and S. Geva. "A Distributed Connectionist Representation for Concept Structures." *Program of the Ninth Annual Conference of the Cognitive Science Society.* Seattle, WA: Cognitive Science Society, 1987.

[89] L. G. Valiant. "A Theory of the Learnable." *Communications of ACM* 27, No. 11 (1984): 1134–1142.

[90] D. E. Van den Bout and T. K. Miller. "A Traveling Salesman Objective Function That Works." *IEEE International Conference on Neural Networks Vol. II.* San Diego, CA: IEEE TAB Neural Network Conference, 1988.

[91] V. Vemuri. *Neural Networks: Artificial Neural networks: Theoretical Concepts*, Los Angeles, CA: Computer Society Press, 1988.

[92] Y. Wang, J. B. Cruz, Jr., and J. H. Mulligan, Jr. "An Enhanced Bidirectional Associative Memory." *IEEE ICNN International Joint Conference on Neural Networks Vol. I.* San Diego, CA: IEEE TAB Neural Network Conference, 1989.

[93] P. D. Wasserman. *Neural Computing: Theory and Practice*, New York, NY: Van Nostrand Reinhold, 1989.

[94] S. H. Weber. "A Connectionist Modèl of Conceptual Representation." *IEEE ICNN International Joint Conference on Neural Networks Vol. I.* San Diego, CA: IEEE TAB Neural Network Conference, 1989.

[95] G. Weisbuch. "Scaling Laws for the Attractors of Hopfield Networks." *Journal of Physique Lett.* 46 (1985): L623–L630.

[96] G. Weisbuch and D. D'Humieres. "Determining the Dynamic Landscape of Hopfield Networks." in *NATO ASI Series, Vol. F20, Disordered Systems and Biological Organization*, pp. 187–191, Berlin, Heidelberg: Springer-Verlag, 1986.

[97] G. V. Wilson and G. S. Pawley. "On the Stability of the Travelling Salesman Problem Algorithm of Hopfield and Tank." *Biological Cybernetics* 58 (1988): 63–70.

7. APPENDIX A: Brown's Martingale Central Limit Theorem and Cramer-Wold Device Theorem

Definition : Let's define S_n as $\sum_{k=1}^{n} Z_k$. Then $\{S_n\}_{n=1}^{\infty}$ is *a Martingale with respect to (w.r.t)* $\{\vec{V_n}\}_{n=1}^{\infty}$ if $\mathbf{E}|S_n| < \infty$ for all n and $\mathbf{E}\left(S_{n+1}|\vec{V_1}, \ldots, \vec{V_n}\right) = S_n$, where $\{Z_n\}$ is a sequence of random variables and $\{\vec{V_n}\}_{n=1}^{\infty}$ is a sequence of random vectors such that $\vec{V_n} = [X_n^1, \ldots, X_n^m]$ for a fixed m.

Brown's Martingale Central Limit Theorem : Assume that $\{S_n\}_{n=1}^{\infty}$ is a Martingale *w.r.t* $\{\vec{V_n}\}_{n=1}^{\infty}$. Let $\delta_n^2 = \mathbf{E}\left(Z_n^2|\vec{V_1}, \ldots, \vec{V_{n-1}}\right)$, $U_n^2 = \sum_{i=1}^{n} \delta_i^2$, and $\&_n^2 = \mathbf{E}(U_n^2)$. If (1) $\frac{U_n^2}{\&_n^2} \to 1$ in probability, and (2) $\lim_{n\to\infty} \frac{\sum_{k=1}^{n} \mathbf{E}|Z_k|^4}{\&_n^4} = 0$, then $\frac{S_n}{\&_n} \Rightarrow \mathbf{N}(0,1)$.

Theorem : Under Assumption 3.1, approximately

$$N_i^s(n,m) \sim \mathbf{N}(0, \frac{(m-1)(n-1)}{2}).$$

This is the detailed proof of Proposition 4.4.

Proof: Fix m and let (1) $Z_n = \sum_{s'=2}^{m} X_1^{s'} \cdot X_n^{s'} \cdot v_n^1$, (2) $S_n = \sum_{k=2}^{n} Z_k$, (3) $\vec{V_n} = [X_n^1, \ldots, X_n^m]$, (4) $\delta_n^2 = \mathbf{E}(Z_n^2|\vec{V_2}, \ldots, \vec{V_{n-1}})$, (5) $U_n^2 = \sum_{i=1}^{n} \delta_i^2$, and (6) $\&_n^2 = \mathbf{E}(U_n^2)$ for $n = 2, 3, \ldots$. Because $\mathbf{E}(S_n) < \infty$, and

$$\begin{aligned}
&\mathbf{E}(S_{n+1}|\vec{V}_1,\ldots,\vec{V}_n) \\
&= \sum_{j=2}^{n}\sum_{s'=2}^{m} X_1^{s'}\cdot X_j^{s'}\cdot v_j^1 + X_1^{s'}\,\mathbf{E}(\sum_{s'=2}^{m} X_{n+1}^{s'}\cdot v_{n+1}^1|[X_j^1,\ldots,X_j^m], j=1..n) \\
&= \sum_{j=2}^{n}\sum_{s'=2}^{m} X_1^{s'}\cdot X_j^{s'}\cdot v_j^1 = S_n \text{ for } n=2,3,\ldots,
\end{aligned}$$

$\{S_n\}_{n=2}^{\infty}$ is *a Martingale w.r.t* $\{\vec{V}_n\}_{n=2}^{\infty}$. And,

$$\begin{aligned}
\delta_n^2 &= \mathbf{E}(Z_n^2|[X_j^1,\ldots,X_j^m], j=2..n-1) \\
&= \sum_{s_1=2}^{m}\sum_{s_2=2}^{m}\mathbf{E}(X_1^{s1}\cdot X_1^{s2}\cdot X_n^{s1}\cdot X_n^{s2}\cdot (v_n^1)^2|[X_j^1,\ldots,X_j^m], j=2..n-1) \\
&= \sum_{s=2}^{m}(X_1^s)^2\cdot 1/2 = (m-1)/2, \text{ also,} \\
U_n^2 &= \sum_{k=2}^{n}\delta_k^2 = (n-1)(m-1)/2. \text{ Therefore,} \\
\&_n^2 &= \mathbf{E}(U_n^2) = (n-1)(m-1)/2, \text{ and} \\
\frac{U_n^2}{\&_n^2} &= 1. \text{ In addition to that}
\end{aligned}$$

$$\lim_{n\to\infty}\frac{n\mathbf{E}|Z_2|^4}{(n-1)^2(m-1)^2/4} = 0.$$

Hence, by Martingale Central Limit Theorem,

$$\frac{S_n}{\&_n} = \frac{\Sigma_{j=2}^{n}\Sigma_{s=2}^{m}X_1^s\cdot X_j^s\cdot v_j^1}{\sqrt{(n-1)(m-1)/2}} = \frac{N_i^s(n,m)}{\sqrt{(n-1)(m-1)/2}} \Rightarrow \mathbf{N}(0,1). \quad \square$$

Cramer-Wold Device Theorem : For random vectors $X_n = (X_{n1}, X_{n2}, \ldots, X_{nk})$ and $Y = (Y_1, Y_2, \ldots, Y_k)$, a necessary and sufficient condition for $X_n \longrightarrow Y$ is that $\sum_{u=1}^{k} t_u \cdot X_{nu} \longrightarrow \sum_{u=1}^{k} t_u \cdot Y_u$ for each $(t_1, \ldots, t_k)$ in $\mathbf{R}^k$.

8. APPENDIX B: Pseudo Code for the Algorithm of Weisbuch's Simulation

PROCEDURE: SIMULATION

Input:

- **NumLoop** {* test cycles for checking probability *}
- **NumNeurons** {* n *}
- **GivenP** {* given probability *}

Output:

- m {* the maximum number of patterns which can be saved with the probability GivenP in a n-neuron Hopfield neural network *}

Algorithm:

$NumofOK = 0$;

L: Repeat NumLoop times

- Generate t random patterns;
- Set the weights according to Hebbian Learning Rule;
- **If** all of the t patterns satisfy the invariant conditions (**IC 1 and 2**) **Then** $NumofOK = NumofOK + 1$;

Probability Checking

- Probability of storing t random patterns:

$$\mathbf{P} = \frac{NumofOK}{NumLoop} \times 100;$$

- **If** $\mathbf{P} \geq$ GivenP
  Then

  - $t = t + 1$;
  - Go To **L**;

  Else

  - **Return**$(m = t)$;
  - **STOP.**

9. APPENDIX C: A Sample Database and Sample Runs of ILDBMN

DATABASE JETS AND SHARKS

NAME	GANG	AGE	EDUCATION	MARRIAGE	OCCUPATION
Pete	Jet	20	H.S.	Sing	Bookie
Fred	Jet	20	H.S	Sing	Pusher
Gene	Jet	20	COL	Sing	Pusher
Phil	Shark	30	COL	Marr	pusher
Don	Shark	30	COL	Marr	Burglar
Ned	Shark	30	COL	Marr	Bookie

SAMPLE RUNS OF ILDBMN

```
Script started on Tue Jun 26 10:26:33 1990
> prolog -h 1000 -g 1000 -l 500
C-Prolog version 1.5
| ?- [userio, userinterface, binder, neuroinference, weights].
userio consulted 19744 bytes 1.98333 sec.
userinterface consulted 5004 bytes 1.25 sec.
binder consulted 1408 bytes 0.383334 sec.
neuroinference consulted 8200 bytes 1.95 sec.
weights consulted 17320 bytes 4.83333 sec.

yes
```

```
| ?- ildbmn.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 ILDBMN is an experimental database system
 using Macro-Neuron Based Inference Engine. And
 general concepts are automatically generated
 from the given attributes by another neural
 network, what we call Bidirectional Associative
 Memory (BAM).+++++++++++++++++++++++++++++++++++
 The system consists of: USERIO, USERINTERFACE,
 BINDER, NEUROINFERENCE, WEIGHTS.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                  .... type any key to continue..

========================================
     SELECT THE CORRESPONDING NUMBER
       FOR YOUR OPTION

     --------------------------------
     1. New-Transaction
     2. Continue-Transaction
     3. Termination
========================================
     Option Number :  1.

     /*<< We want to refresh STM >>*/

========================================
     [OPTIONS FOR INPUT ATTRIBUTES]

     --------------------------------
     For known data: type the data
     For unknown data : type  no
========================================
     Name : no.
     Gang : 'Jet'.
     Age : no.
     Education : no.
     Marriage : no.
     Occupation : no.
```

```
     /*<< We have only the information >>*/
     /*<<          Gang : Jet           >>*/

=======================================
[OPTIONS FOR INTERESTING ACCESS TO STM]
++++++++++ type  y  or n ++++++++++++
=======================================
     Access to Attributes : y.
     Access to Concepts : n.

     /*<< We want to access attributes >>*/
     /*<< in a stabilized STM          >>*/

=======================================
[INTERESTING ATTRIBUTES FROM STM]
TYPE THE CORRESPONDING NUMBERS
FOR YOUR OPTION
---------------------------------------
  1. Name  2.Gang  3. Age  4.Education
  5. Marriage  6. Occupation
=======================================
INPUT
|: 1.

/*<< Specifically, who are the member of Jet >>*/

        .....please, wait for a moment..
   /*<< Call NEUROINFERENCE >>*/


    Block Cycles for Attributes:  5
    /*<< Stabilized in 5 block cycles >>*/

===========[ATTRIBUTES]============
          name  :  Pete
          name  :  Fred
          name  :  Gene

    /*<< They are Pete, Fred, and Gene >>*/
```

.... type any key to continue..

```
========================================
     SELECT THE CORRESPONDING NUMBER
       FOR YOUR OPTION
     -------------------------------
     1. New-Transaction
     2. Continue-Transaction
     3. Termination
========================================
     Option Number :  2.

     /*<< We don't want to refresh STM >>*/

========================================
[OPTIONS FOR INTERESTING ACCESS TO STM]
++++++++++ type  y  or n ++++++++++++
========================================
     Access to Attributes : n.
     Access to Concepts : y.

     /*<< We want to access generated concepts >>*/
     /*<< from the given input: Gang-Jet       >>*/

        .....please, wait for a moment..
   /*<< Call NEUROINFERENCE >>*/

    Block Cycles for Concepts:  1
    /*<< Stabilized in 1 block cycle >>*/

++++++++++++[CONCEPTS]++++++++++++
          age  :  20
          marriage  :  Sing

     /* Generally, the member of Jet are >>*/
     /* 20 years old and are single      >>*/
```

```
                    .... type any key to continue..


=======================================
     SELECT THE CORRESPONDING NUMBER
       FOR YOUR OPTION
     --------------------------------
     1. New-Transaction
     2. Continue-Transaction
     3. Termination
=======================================
     Option Number :  3. /*<< EXIT ILDBMN >>*/

[ execution aborted ]

| ?- ^D
[ Prolog execution halted ]
> exit
```